AD A063593
AD-E400 230
RIA-78-U594

TECHNICAL REPORT ARPAD-SP-78001

GENMOD

USER MANUAL FOR GENERALIZED

PRODUCTION LINE MODELING ROUTINE

EDWARD E. LONIEWSKI

AUGUST 1978

U.S. ARMY ARRADCOM

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND

PRODUCT ASSURANCE DIRECTORATE

DOVER, NEW JERSEY

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ARPAD-SP-78001	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (*and Subtitle*) GENMOD - User Manual for Generalized Production Line Modeling Routine		5. TYPE OF REPORT & PERIOD COVERED FINAL
		6. PERFORMING ORG. REPORT NUMBER DRDAR-QAS-1-78
7. AUTHOR(*s*) Edward E. Loniewski		8. CONTRACT OR GRANT NUMBER(*s*)
9. PERFORMING ORGANIZATION NAME AND ADDRESS ARRADCOM DRDAR-QAS Dover NJ 07801		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS ARRADCOM DRDAR-TSS Dover NJ 07801		12. REPORT DATE August 1978
		13. NUMBER OF PAGES 162
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)		15. SECURITY CLASS. (*of this report*) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Computer simulation
Computer modeling
Production lines

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

This report represents a thorough introduction to the modeling and coding techniques needed to effectively use the computer program called GENMOD to analyze the design and performance characteristics of automated production lines. The program was developed by the Product Assurance Directorate, ARRADCOM, and has been used by various organizations to successfully model a wide variety of lines.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

20. Abstract. (Continued)

The intent of this manual is four fold:

1. to explain the general functioning of the program

2. to provide detailed information on each of the building blocks available.

3. to present numerous examples of how the blocks are combined to construct production lines

4. to alert the user to various special features of the program which might prove valuable.

TABLE OF CONTENTS

ABSTRACT

This report represents a thorough introduction to the modeling and coding techniques needed to effectively use the computer program called GENMOD to analyze the design and performance characteristics of automated production lines. The program was developed by the Product Assurance Directorate, ARRADCOM, and has been used by various organizations to successfully model a wide variety of lines.

The intent of this manual is four-fold:

1. to explain the general functioning of the program

2. to provide detailed information on each of the building blocks available

3. to present numerous examples of how the blocks are combined to construct production lines

4. to alert the user to various special features of the program which might prove valuable.

ACKNOWLEDGEMENT

When this program was first developed it was done so from the standpoint of satisfying the requirements which existed in my office at the time. If the program had continued to be used solely by that group for the same class of problems it would undoubtedly have undergone very little modification and expansion. Once word got out, however, and the program began to be used by other groups, a great number of changes were made and new features added as each new user requested special routines or suggested improvements to handle his particular line. I am especially indebted to Mr. Fernando Nin of Harry Diamond Labs for the long hours he put in debugging the many coding changes I made to implement the worthwhile modifications he requested. Without such outside help I would never have incorporated into the program all the features that exist now.

I would also like to give a big hand to Mrs. Dottie Landiak who helped so much in putting this book together. Her skill and patience were the only things that got us through the many months this manual consumed.

Edward E. Loniewski

This page intentionally left blank.

1. INTRODUCTION

1.1 GENMOD Rationale

This first section is designed solely to present the basic ideas behind the GENMOD system. Many new concepts and definitions will be introduced which at first may confound the reader. But, be patient. No actual modeling will be done here. That will come later, after all major points have been covered at least once. Here we will be concerned only with what sorts of information are needed to simulate a production line and how GENMOD goes about tying these pieces of information together.

Wherever possible, figures and examples are used to illustrate a point. They are deliberately bare and simple, not to frustrate the reader, but to show that all cases, no matter how fancy and sophisticated, can be reduced to a basic, invariant reference point. These basic points must be mastered before tackling the "real world".

General Modeling System (GENMOD) was developed in response to a need to model various automated production lines to study the effects of proposed design changes. At first each line studied was modeled by a separate computer program. It soon was apparent, however, that although each line had its "special" features there was a great deal of repetition among all these programs and a little thought revealed several basic similarities. An attempt was made, then, to write a generalized model program through which a specific line could be analyzed by considering it as data to the program rather than as a separate program in itself. This could be done if a suitable means were developed to:

a. characterize the operation of a particular machine and provide an adequate number of model machines,

b. specify how groups of machines operate with respect to each other,

c. precisely define product flow through the line, and

d. impose rules of operations to lines in general.

It would then be possible to take any production line, code it according to some set of rules, feed these codes as input to a universal program, and perform a detailed analysis as if the program were written for that specific line.

This report shall present all of these generalizations and the reason for their existence, explain how a line can be described using these generalizations, and work through several specific examples.

1.2 Production Line Characterization

For GENMOD's purposes, a production line is characterized as a sequence of distinct independent operations that are performed in turn on an identifiable product. Prior to the first operation, this product is considered as nothing more than a pile of raw material and following the last operation it is a finished product. At any point in between, the product is considered semi-processed. Each operation receives its share of semi-processed material from a specific location and, having done its job, transfers the material to another location. In GENMOD, there is no such thing as a machine handling parts directly to another machine. All material must pass through some non-operating point first. This passage may be virtually instantaneous, but it does indeed take place, not only in GENMOD, but in reality as well.

1.3 Motion Picture Analogy

Perhaps the best aid available to understand the workings of GENMOD is to envision the entire production line being filmed and then each frame of that film being examined carefully. Each such frame would show quantities of material in various stages of production scattered throughout the line and, if some sort of marking scheme was employed, it would also show which machines were up and running and which machines were not. GENMOD does nothing more than to try to take each frame of this film and quantify all the information on it. This analogy will indeed prove helpful later.

Depending upon the amount of detail desired, a line could be filmed at all sorts of speeds, ranging from super slow-motion to high speed time-lapse. Whatever the speed, though, each frame obviously represents some interval of elapsed real time. Why one speed might be selected over another will be discussed later in this text. However, once a certain speed is selected, there is a corresponding time interval covered by each frame.

GENMOD can also run at different speeds, and for each of these speeds there is a corresponding frame time called the basic time interval (BTI). Its proper selection is critical to GENMOD's accuracy and will be covered in depth later.

1.4 Machine Characterization

The basic assumption made by GENMOD in characterizing the operation of a machine is simply this:

> Every machine can be thought of as a black box which, when up, receives parts from one fixed input point, does something to these parts at a fixed rate, and deposits the parts into one fixed output point.

With very few exceptions, this rule is adhered to strictly.

Thus, a machine can be completely described independently of all other machines by specifying five pieces of information:

a. input point

b. output point

c. availability

d. production rate

e. mode of operation

Given this information on all machines it is possible for the program to develop tables for product flow which enable it at any given instant in time to know precisely which machines are up or down, exactly how much semi-processed material is located at each input/output point, and when it will be necessary to stop an operation due to obstructions or shortages on the line.

1.5 Single-Machine Line

Consider for a moment the very simplest line possible, i.e. one consisting of a single machine located between just two material points:

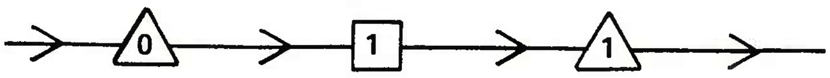


Figure 1

(Throughout this report and all other references to GENMOD I have used the convention that machines are shown as boxes, numbered from 1, and material points as triangles, numbered from 0.)

A basic time interval is selected, representing the smallest period of time for which the program keeps track. For this example, let this interval be one minute. Having made this choice, the characteristics of the machine must be stated in these terms also, e.g. a maximum production rate of 50 parts per minute, a mean time between failures of 360 minutes, etc.

(Please note that many references will be made to minutes. This is only for illustrative purposes. Nothing in GENMOD is geared to any specific time value and the user should not feel that he is either.)

As long as a machine is up and running, it will in every interval of operation attempt to process as many parts as it is capable of handling, subject to the constraints that there are that many parts available in its input point and there is room for that many parts in the output point. Other considerations come into play when there are more machines present, but they are of no interest yet.

In general, the lowest numbered material point is the head of the line, or raw material point, where material is first available to any machine. Likewise, the highest numbered material point is the tail of the line, or final output, where finished parts exit from the line and are no longer of any interest. The user specifies the amount of raw material that will be made available at point 0 every interval and one of the program's basic assumptions is that this amount is never interrupted.

Under certain conditions, it is possible to model more than one raw material point, representing several entries of material or sub-assemblies onto the line. This feature will be discussed in time. For the moment, however, we shall stick to only one such point and it shall always be point 0. In fact, regardless of whether other raw material points are present or not, every GENMOD model must have a point 0 and it is always considered a raw material point.

1.6 Failure/Repair Sampling Distributions

With a line as simple as this, the final output is solely a function of how often the one machine is up, which brings us to the problem of determining when a machine is to fail and how long it is to remain down for each failure. Part of the input for each machine represents availability data in the form of a failure distribution and a repair distribution. Each of these distributions must be fully specified by the appropriate parameters.

At present, two choices are available for failures: normal (Gaussian) or exponential. If the user wishes to characterize failures as normally distributed, a mean and a sigma must be specified. If the exponential distribution is to be used, only the mean is specified.

For repairs, the choice is between log-normal and exponential. Here, for the log-normal, the user again supplies a mean and sigma, using a convention that the sigma is to be expressed in terms of the natural log of the physical values. Again for exponential, only the mean is to be specified, with the sigma value being left blank or zero.

Additional choices for distributions can easily be incorporated.

One of the first things the program does is to determine the time to first failure for each machine. This is done by choosing a random number from the computer's random number generator, equating it to a point on the specified cumulative failure distribution, and calculating the physical value (time) to which this corresponds. As each machine fails and subsequently is repaired, a time to next failure is similarly calculated. Thus, a table of times-to-next-failure for each machine is continually maintained. When the time of simulation reaches the point at which a particular machine is to fail, the status of that machine is changed from "up" to "down", it is removed from the current process flow of the line, and a time to repair is calculated from the repair distribution in a manner identical to failure time determination. After passage of this time, the machine is restored to "up" status and a time-to-next-failure is selected. Associated with every machine on the line, therefore, at all times is a status word (up/down) and an event time (failure/repair).

Since GENMOD, like a movie camera, works in discrete time intervals, all event times (failures and repairs) must be at least one interval long and they are automatically rounded up if ever the numerical calculation results in a time less than one interval. This reality is just one of the considerations that go into selection of the proper BTI. For example, if a mean time to repair (MTTR) is known to be about two minutes and distributed exponentially, then many typical repair times will be less than one minute. To choose a BTI of one minute in such a case will cause biased results, in that these times will be arbitrarily raised to one minute. However, choosing a BTI of 15 seconds, for example, changes the MTTR to eight intervals, rather than two, and the occurrence of times less than one interval is greatly reduced. This is only one of infinitely many BTI's that could be chosen. Further examples later will clarify this situation.

1.7 Double-Machine Line

Several important and interesting concepts can now be introduced if we expand our original example to include one more operation in series with the first machine:

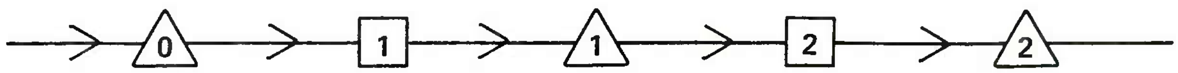


Figure 2

At least four new topics arise here which were of little or no interest in the simpler model:

1. necessity for material point

2. relationship between machine rates

3. upstream/downstream stop times

4. importance of basic time interval

If we consider the horizontal line across the diagram as the production line and if we adhere to our definition of a machine as being a black box that receives its input parts from somewhere and deposits its finished parts somewhere else, we are forced to provide for the existence of material point 1 as being a location on the line between the two machines which serves as the output point of Machine 1 and the input point for Machine 2. Since machines cannot pass parts directly between each other, it is important to note the existence of this point and the significance it holds for the successful modeling of a line.

Depending upon the physical construction of the line, this point can have several interpretations. Where there is an actual length of conveyor between machines, this point can be used for that length of conveyor, especially when there is no need to depict actual travel time along the conveyor. On those lines which have physical storage buffers present, these too can be represented by these points. (In fact from here on, these material points shall be referred to as "buffer" points regardless of whether they represent physical buffers or not, in as much as they exist as potential buffers and must be taken into account.) Proper use of buffer points will become apparent as we proceed.

1.8 Inter-Machine Relationships

In the first example, it was possible to calculate output simply by knowing how long the machine was up and how much raw material was available. Here, this is not as easy because in addition to knowing how long both machines are up, we must also take into account the rates at which the machines operate and the capacity of the buffer point in between. For calculation's sake, let R_1 and R_2 be the rates of the two machines respectively, B_1 be the current contents of the buffer point, and C_1 be the maximum capacity of that point. Then, under the steady state situation (i.e. when all machines are up and running and no buffers are full), three cases exist:

1. $R_1 < R_2$: Machine 2 is forced to operate at rate R_1, the line will run smoothly, and the final output is a function of R_1

2. $R_1 = R_2$: both machines run at the same rate, the line operates smoothly, and the output is again a function of R_1

3. $R_1 > R_2$: a backlog is created and B_1 will increase by $(R_1 - R_2)$ units per interval, continuing for $(C_1 - B_1)/(R_1 - R_2)$ intervals, after which Machine 1 will slow down to R_2; the output is a function of R_2

As long as the steady state situation exists, it is still possible to calculate output by knowing the relationship between rates, as shown above. (This, of course, becomes more difficult as the number of machines increases.) Once machines start to fail, however, it is almost impossible to determine output by hand. Not only must the failures and repairs themselves be considered, but also their relationship to other machines' failures and the contents of the buffers when they occur. It is the ability to consider interrelationships which makes computer simulation so valuable over hand calculations.

When a maximum size is placed upon a buffer point, in an environment of machine failures, there arises the problem of buffers either becoming full (as in Case 3 above), causing stoppage of machines due to lack of room to deposit output, or becoming empty, causing stoppage due to lack of input material. Determining just when a buffer point becomes full or empty is a problem that involves all machines and buffers on the line and not just the machines immediately before and after the buffer in question. For every interval of simulation, GENMOD determines which machines are up or down, what the parts flow is for every machine, and what the expected input and output are for all buffer points. It then smooths this flow, making adjustments to machine handling where necessary before transferring parts. Only then does the program consider whether or not a buffer is full. Statistics are gathered by the program whenever it is necessary to stop a machine because of a full output point (<u>downstream stop time</u>) or because of an empty input point (<u>upstream stop time</u>).

1.9 <u>Basic Time Interval (BTI) Considerations</u>

With as few as these two machines in our model it is possible to start discussing the importance of selection of the proper basic time interval (BTI). Several realities of the program should be borne in mind when considering this choice:

a. Whatever the interval, all machines are assumed to complete their action within this time, to include transferring parts from the input point to the output point; thus, if an interval of 4.5 minutes is selected and a rate of 18.0 is stated for a given machine, then each step through the line will represent 4.5 minutes of real time and this machine is considered as completely processing 18.0 parts during this time, subject of course to up/down and buffer constraints; note that this combination is equivalent to a rate of 4 parts per minute (PPM), but this may not be the correct interval to choose.

b. Remember that all speeds and event times <u>must</u> be specified in terms of the BTI; furthermore, the BTI is not necessarily related to the so-called cycle time of any machine, although this certainly has a bearing on the choice.

c. Once a machine has processed its parts and deposited them in its output point, those parts will <u>immediately</u> be available to the next machine at the start of the next interval; the consequences of this rule are two-fold:

(1) Parts are transferred completely from buffer to buffer in one interval, thus making total travel time on the line equal to the number of series steps between buffers

times the length of the interval; in figure 2, it takes one interval to travel from 0 to 1 and one interval from 1 to 2, for a total of two intervals; with a BTI of 1 minute this means 2 minutes total residence time; an interval of 4.5 minutes means residence time of 9 minutes and so on; it is clear, then, that the choice of BTI is intimately related to residence time on the line, to conveyor speed, and to the distance from one buffer to the next.

(2) If an input buffer is empty at the beginning of an interval, all machines being fed by that point will be stopped for the entire interval, even if (in reality) that buffer were to become partially filled during the interval; in figure 2, starting out with empty buffers, in interval 1 only machine 1 will work, in interval 2 both machines will work, and so on; thus, if a model contains n series steps, machines in the final step will not begin to work until (at least) the nth interval.

d. GENMOD happens to be machine oriented, taking buffers into consideration, unlike other programs which are just the opposite; thus, in every interval it looks at every machine to determine process flow; the most successful models are those which have the machines numbered (generally) from top-to-bottom and left-to-right; this is important to remember in view of the other consequences of the BTI; a line which is numbered in, say, a random fashion will probably work, but the timing will almost definitely be disrupted.

Without yet having introduced delay elements, batch operations, or dummy machines, it is difficult to give a precise rule for determining BTI. At this point, let the following definition suffice:

On a line where all stations (not machines) have approximately the same process duration and all series buffer points are approximately equi-distant, then BTI should be the total residence time divided by the number of series stations.

For example, a line consisting of 14 equally important steps of about the same duration and which takes 38 minutes for the first finished part to reach the output from a dead start, then BTI = 38/14 = 2.71 minutes. Remember, the program does not care what space of real time is represented by the BTI, so long as all rates are given as parts per basic time interval (PPBTI) and all other timing aspects are consistently stated, such as failure, repair, and delay times.

Little difficulty will be encountered with BTI on lines which have a relatively high volume, i.e. several parts per minute rather than several minutes per part. This is so because the length of time any one part is in any one machine is so short that minor variations in cycle time between machines will hardly be noticed. There will be trouble, however, when the cycle times are so long that they start to approach a proposed BTI. In general, either BTI should be longer than the residence time of any machine on the line or the machines should be defined as requiring several BTI to complete their function.

1.10 Parallel Stations and Branches

The above paragraphs made reference to such things as stations, series steps, delay elements, and other terms which may or may not be familiar but which now will be discussed in detail. To do so, let us consider two more relatively simple models, using a vertical format, which shall be adhered to from now on. These figures will introduce some new concepts and some subtle problems.

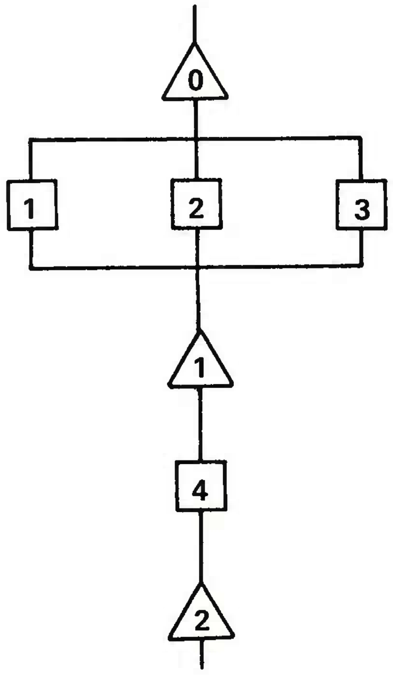


Figure 3

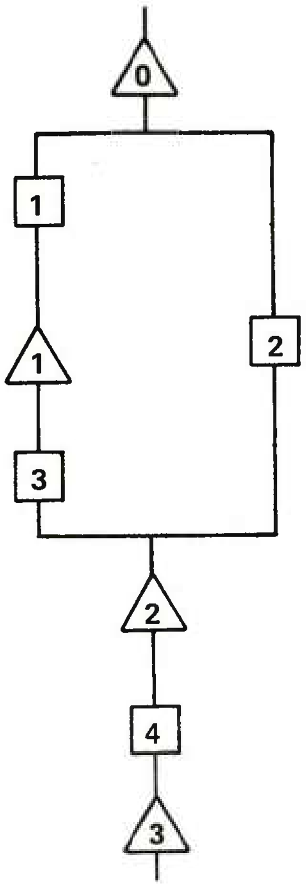


Figure 4

In figure 3 we have for the first time several machines in a parallel station, i.e. a group of machines performing some operation on the same group of parts coming from the same input point and going to the same output point. Figure 4 introduces parallel branches on a line, i.e. strings of machines in series which begin at a common buffer point and come together again at a common point. A station can be thought of as being several branches of one machine each, just as Machine 2 in Figure 4 can be considered a branch by itself. But, normally, a branch consists of two or more machines or stations in series, i.e. following each other both physically and logically.

Both figures represent workable lines and are numbered in accordance with the conventions of GENMOD. Notably, the machines are numbered consecutively from top-to-bottom and left-to-right, every machine has one and only one input and one and only one output, and every path between machines not in the same station goes through a buffer point.

Machines 1, 2, and 3 of Figure 3 represent one of two situations: either they are of different function or else they work together to do the same job. When different, they may even have different rates and probably different failure and repair distributions. Such an arrangement is not common. But, it can exist and GENMOD can model it. The more usual situation is for the machines to be of the same design, performing the same operation at equal rates, subject to the same failure and repair patterns, and sharing the same input more or less equally.

1.11 Machine Categories

To accomodate the fact that there may be multiple machines of the same design at various places on a line, GENMOD provides for identifying not only individual machines, but also classes of machines as well. These classes are called categories and are defined by the user, as he chooses, to classify machines according to their function, e.g. grinders, borers, polishers, inspectors, etc. This grouping is for identification and statistical purposes only. It in no way affects the logical running of the line. Within each category, there can be any number of machines, either close to each other or widely spaced, and they are tagged as being the 1st, 2nd, etc. machine of that category. Such enumeration is totally independent of the numbering of machines on the flow chart, which should always be done first. Breaking the machines down into categories can be done afterwards. In fact, there is no reason why every machine can't be assigned a category of its own, in which case there will be n machines with n categories of one machine each.

These categories are defined simply by reading in a list of titles for them, the index number of each title then being used to tie machines with their respective category. At the end of a simulation run, statistics will be printed on the individual machines and then on the categories also.

On the assumption that the first three machines of Figure 3 are of the same design and same category, then they can either be working in unison with each other or in competition with each other. The former situation exists when the conveyor or whatever which moves parts out of Buffer 0 is designed to distribute these parts more or less equally to all three. In the latter case, the conveyor is designed to consistently feed Machine 1 first till its capacity is reached, then Machine 2 second, and so on until the input point is empty. GENMOD allows for both such designs as well as several other more complex arrangements. All these combinations will be discussed in detail in the section on building blocks.

1.12 Spacing Considerations

The line in Figure 4 presents a new problem in that the left-hand branch consists of two operations, thus requiring two time intervals, while the right-hand branch consists of only a single operation requiring a single interval. If this is indeed the way the line works, then this model is adequate. If, however, Machine 2 actually takes as long as the other two machines together to process its amount of material, then something else must be done to the model. Three such possible modifications, which will also be discussed in depth later, are:

a. Delay the output from Machine 2 for one interval so that it arrives at Buffer 2 at the same time as the output from the other branch

b. Insert a so-called dummy machine in series behind Machine 2 so that both branches are of the same length; such dummy operations do no real processing of parts but are extremely useful in line balancing problems

c. Define Machine 2 as a batch type machine which takes two intervals to process material; in this method, care must be taken to define batch size properly.

In both figures, the output from the multiple operations is deposited in one buffer with no regard to which machine last handled which part. In other words, the finished parts from machines 1, 2 and 3 in Figure 3 all go into Buffer 1 where their identity is lost. If for some reason it is important to get detailed information on parts flow at this level, the user has two options. He can insert a dummy machine behind each of the three machines, thus creating three new buffer points. Detailed statistics will then be available on these points at the expense, however, of having added one more time interval to the line's normal residence time. A second choice is to simply make use of the statistics already gathered by the program on machine utilization, namely total parts handled by each machine for the length of the run (printed automatically) and parts handled by each machine over a given period of time (user specified). These values will usually be enough to gather the information sought. There will be further discussion of this problem when dummy machines are described.

We are now ready to move on to the subject of discussing the types of components available with which to construct a model.

This page intentionally left blank.

2. COMPONENT TYPES

2.1 Machine Specification

In general a machine is completely described to GENMOD by supplying 13 pieces of information:

a. machine number, as on flow chart

b. category number, as on category list

c. sequence number within category

d. input buffer point

e. output buffer point

f. maximum rate, in parts-per-basic-time-interval

g. meantime between failure (MTBF)

h. sigma on TBF (or 0.0 for exponential)

i. meantime to repair (MTTR)

j. sigma on TTR (or 0.0 for exponential)

} in terms of BTI

k. defect rate, from 0.0 to 1.0

l. type code, as herein described

m. auxiliary information (as may be required)

Emphasis will be placed here upon items l and m and their relationship with other values, as well as what results they will generate.

All of these above items are read from a single data card. Since reference is made in the following sections to specific data values on the card, we would do well to briefly examine now the required format for these cards. Figure 5 shows the card layout for the 13 data fields (a thru m), their code word, their FORTRAN format, and several typical (but unrelated) sample cards.

This page intentionally left blank.

FIELD	a	b	c	d	e	f	g	h	i	j	k	l	m	
CODE	MACH	CAT	SEQ	IN	OUT	PPM	MTBF	STBF	MTTR	STTR	DEF	TYPE	AUX	----
FORMAT	I3	I3	I3	I3	I3	F8	F8	F8	F8	F8	F8	I3	I3	----
COLUMNS	1-3	4-6	7-9	10-12	13-15	16-23	24-31	32-39	40-47	48-55	56-63	64-66	67-69	70-80
	1	1	1	0	1	14.20	45.00	5.000	3.000	0.110	0.000	1	0	
	13	5	3	14	16	180.0	60.00	10.00	0.000	0.000	0.001	-3	2	
	20	10	1	10	999	0.000	96.40	8.320	0.000	0.000	0.000	5	0	
	32	0	0	0	0	0.000	0.000	0.000	0.000	0.000	0.000	99	0	

Figure 5

This page intentionally left blank.

No attempt will be made in this section yet to actually put together a model. Rather, we will just discuss the types of building blocks, and their variations, available.

2.2 Type 0: Competitive Operations

The basic and most commonly used machine or operation in GENMOD's inventory is a Type 0. An element of this type receives its inputs from one source location, processes this material at a fixed maximum speed (subject to the quantity of material available), and transfers its processed material to one output point, all in an environment of sharing inputs with other similar machines in the same station.

When two or more Type 0 components are placed in parallel between the same pair of input/output buffers, these components will vie equally for the input material (subject to the up/down status of each machine at that instant in time), i.e. the up machines in a Type 0 station will be made to share the input equally. The following example will clarify this convention:

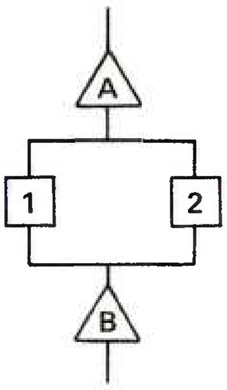


Figure 6

Suppose Machines 1 and 2 have identical characteristics and can process 60 parts per minute each at top speed under ideal conditions and suppose that at the start of a particular minute there are 90 parts available at Point A, with Point B having virtually unlimited capacity.

Then if both machines are up and running for the full minute, each will process 45 parts, Point A will be emptied and Point B will receive 90 parts. If either machine is down for the entire minute while the other is up, the up machine will process 60 parts, Point A will have 30 parts remaining, and Point B will receive 60. Varying results will occur in those situations when one or both of the machines either fail(s) or come(s) back on line during the minute in question. Then the number of parts handled depends upon when during that interval the failure/repair occurred.

These machines can be given the ability to throw a certain percentage of parts off the line, simulating the removal of defective material. Note that this is not equivalent to the manufacture of defective material, unless the machine in question also has the ability to detect these defects and discard them. In other words, GENMOD will account for defective material, but only at the points where this material is detected, just as in real life. Thus, defect removal is associated with an inspection process. Machines which have the capability of detecting and removing bad parts can be made to do so by inputting some value for "DEFECT RATE". This value is sampled from every interval and the appropriate number of parts are removed from the line.

In addition, Type 0 machines can utilize the delay option to be discussed later.

2.3 Type 1: Domineering Operations

The second most commonly used building block in GENMOD is the Type 1. Such a component operates in exactly the same way as a Type 0 with one major exception: when more than one element is present between a pair of nodes, the elements do not share the inputs, but rather they hog the inputs on a first-come-first-served basis, i.e. the first machine to operate will process as many parts as it can before letting the second and subsequent machines handle what remains in turn. Referring to Figure 6 again, if these components were Type 1 rather than Type 0 they would function as follows: If both machines are up, Machine 1 will handle 60 and Machine 2 will handle 30 in every interval. Only if Machine 1 goes down will Machine 2 ever get to process 60 parts.

This type component has two main uses:

a. When there is only one non-specialized machine present between two nodes, designating it as a Type 1 component will result in quicker running times than designating it as a Type 0 component, since the program would not be forced to search first for competing machines that do not exist; there is, of course, no difference in simulation results between calling these machines Type 1 or Type 0, just a difference in running time.

b. When a station logically consists of a prime machine with backup(s), whose purpose is to be called upon only if the first machine is down or if the work load demands it, designating all machines as Type 1 will automatically give the bulk of the work to the prime machine, with only the excess going to the succeeding machines as long as the prime is up.

Type 1 machines can also use the inspection and delay options.

2.4 Types ± 2: Alternating/Sequential Operations

Careful thought about the preceeding two component types will reveal that for a given amount of input material, one of three things will happen: 1) all machines will work on equal shares, 2) only one machine will work with all others idle, or 3) all will work with unequal shares. But what about the situation where the machines take turns operating in some sort of definite sequence, e.g. the first machine operates in the first interval, the second in the second, and so on? Using a Type 0 would force all machines to work every minute while Type 1 would always make machine one function in preference to any other machines. This next component, Type 2, will handle this condition.

Type 2 machines can be thought of as alternating or sequential machines. They exist as a set of two or more elements between the same pair of input/output points and are designed such that only one will operate in a given interval, regardless of how many are in the station. They can be used to represent operations which handle parts or material in some sort of batch or pulsating fashion rather than in continuous fashion as well as machines which must be characterized as taking more than one basic interval to completely process their material. Examples of such operations might be:

a. equipment that operates in one interval and gets recalibrated in the second interval,

b. chemical operations which fill up with material and take several intervals to process,

c. machines which do work every interval but only receive input in discrete chunks at some fixed period rather than a smaller quantity in all intervals.

There are two sub-varieties of this component available. They both operate machines sequentially with only one machine handling all usable inputs. But they differ as to what transpires when a downed machine is encountered in that sequencing, thusly:

+2 = if all machines are up, the first will work in the first interval, the second in the second interval, and so on, with operation passing to the next machine (or returning to the first machine) each interval. If any machine is already down when it is its turn to work, that machine is skipped, the sequencing is automatically advanced, and the next machine in line is operated.

-2 = sequencing is not interrupted to accommodate a downed machine. If any machine is actually down when it is its turn to operate, all production for that interval is lost, as no other machines will be allowed to interrupt their operation to take up the slack. The only exception to this rule comes about if the station contains more machines than are necessary for ideal operation, i.e. in the presence of a true redundant machine. Such a machine would be signaled to operate in this instance.

A couple of examples will clarify use of Type ±2 machines. Consider the following station consisting of just three machines:

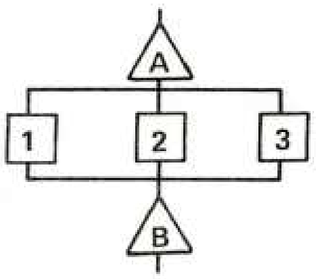


Figure 7

Each machine works on 30 parts at a time, but takes three intervals to do so. The three machines are to receive all 30 parts at once and they are to work one at a time, in turn.

Suppose Point A initially contains 180 parts and Point B is empty. Then the following table shows the number of parts each machine would handle during a 6-interval period if the machines were all coded as Type 0, Type 1, or Type 2. It is assumed that all machines are up for all six intervals.

Interval	Type 0					Type 1					Type ±2				
	Machine			Point		Machine			Point		Machine			Point	
	1	2	3	A	B	1	2	3	A	B	1	2	3	A	B
0	0	0	0	180	0	0	0	0	180	0	0	0	0	180	0
1	10	10	10	150	30	30	0	0	150	30	30	0	0	150	30
2	10	10	10	120	60	30	0	0	120	60	0	30	0	120	60
3	10	10	10	90	90	30	0	0	90	90	0	0	30	90	90
4	10	10	10	60	120	30	0	0	60	120	30	0	0	60	120
5	10	10	10	30	150	30	0	0	30	150	0	30	0	30	150
6	10	10	10	0	180	30	0	0	0	180	0	0	30	0	180

Table 1

Note that the flow into and out of the two buffer points is exactly the same in all cases, with only the parts handling differing.

Now suppose that Machine 1 were to fail at precisely the end of the second interval and remains down for the rest of the study. Table 2 shows the parts flow in this instance.

Note that a mere three machines and two buffers can produce four distinctly different patterns of operation depending upon their usage. In one case, there is even a different output. According to the description accompanying Figure 7, only the last case, Type -2, satisfies the required pattern of operation.

In both varieties of Type 2, if a machine was up when it was its turn to operate, but fails during that interval, then no other machine is given a chance to work, and that production time is lost. No machine will ever operate in consecutive intervals unless all other machines in the group are down at the start of that interval. A downed machine coming back on line in a given interval will not influence the sequencing, as the signal to operate a particular machine is made at the start of the interval before the repair is effected. The repaired machine then resumes its position in line. In all cases, the one machine that is operating in a particular interval will process the material in the input buffer to the extent of its maximum speed (or the maximum amount in the buffer, whichever is less). If there is an excess of material in the input point, it will remain there, as no other machine in the group will be called into action. The program at all times knows which was the last machine that operated.

These two sub-types are differentiated for coding pruposes by simply placing either 2 or -2 in the TYPE field of the card. Furthermore, for Type -2, the number of intervals re-

	TYPE 0					TYPE 1					TYPE +2					TYPE −2				
	MACHINE			POINT		MACHINE			POINT		MACHINE			POINT		MACHINE			POINT	
INT	1	2	3	A	B	1	2	3	A	B	1	2	3	A	B	1	2	3	A	B
0	0	0	0	180	0	0	0	0	180	0	0	0	0	180	0	0	0	0	180	0
1	10	10	10	150	30	30	0	0	150	30	30	0	0	150	30	30	0	0	150	30
2	10	10	10	120	60	30	0	0	120	60	0	30	0	120	60	0	30	0	120	60
3	0	15	15	90	90	0	30	0	90	90	0	0	30	90	90	0	0	30	90	90
4	0	15	15	60	120	0	30	0	60	120	0	30	0	60	120	0	0	0	90	90
5	0	15	15	30	150	0	30	0	30	150	0	0	30	30	150	0	30	0	60	120

TABLE 2

quired for complete processing is placed in the AUX field. Note carefully that this is just one use of the AUX field. The number of Type -2 machines in the station must of course be greater than or equal to the required process time in intervals.

A further comment will be made about -2 machines later when Type 4, batch operation, is introduced.

Type 2 machines can utilize the inspection option, but they cannot directly use the delay option.

2.5 Types ±3: Matchers/Assemblers

The machines described so far all take their inputs from a single point, perform one logical operation on the parts, and deposit the outcome in a single location. There are some operations, however, which work simultaneously on parts from more than one input source, especially if they are sub-assemblies of different types that are combined into a larger assembly.

Consider the two versions of essentially the same line shown in figures 8a and 8b below.

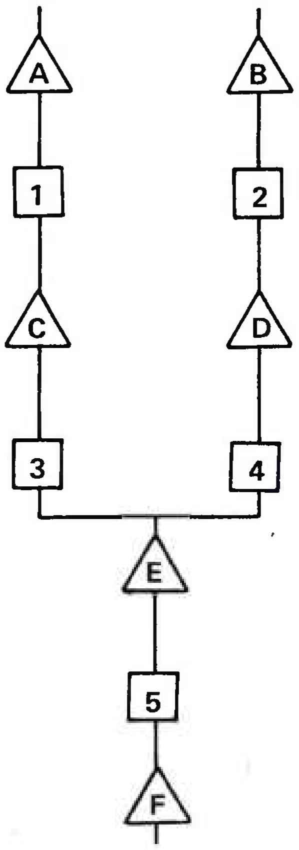


Figure 8a

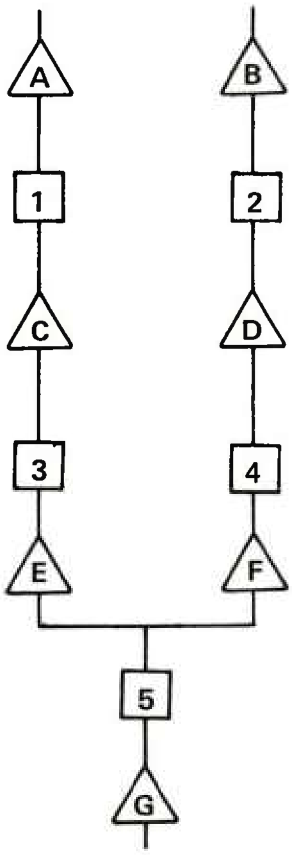


Figure 8b

Both are intended to use Machine 5 to work on the combined outputs of the two branches of the line. In one case, the outputs are dumped simultaneously from Machines 3 and 4 into a common buffer, Point E, from which Machine 5 draws freely. The second version attempts to keep the outputs separate by depositing them in different buffers, Points E and F, and making Machine 5 draw from both simultaneously. Unless there is some design reason for keeping the two outputs separate, only the first version should be attempted with GENMOD, as the second violates the principle that every machine must have one and only one input and one and only one output. On the other hand, if the second version is meant to represent two separate lines of sub-modules which are assembled or matched by Machine 5, then that model is correct, but a new type component must be introduced.

This Type 3 component is designed to act upon a specific set of input buffers only if there are parts present in all such buffers. For all intents and purposes, it continues drawing one part from each buffer until one of the buffers runs out of parts. In this sense it can be thought of as a matching machine in that it will not pass a part from any input buffer to an output buffer unless there are matching parts in all the other input buffers. The number of parts handled then in any interval is the minimum contained on the set of input buffers and this value is further constrained if the maximum rated speed of the machine is smaller.

Having matched a bunch of parts from two or more input buffers, this machine can do one of two things with these parts:

a. pass all of them to the output buffer as individuals,

b. pass them as an equivalent number of higher assemblies.

The first represents a simple matching process, while the second represents an assembly process by which the number of assemblies passed to the output is the same as the number of sub-assemblies pulled from any one input buffer.

For example, suppose this machine is to match parts from three input buffers at a maximum rate of 15 parts per interval from each buffer. In a particular interval, suppose the contents of these buffers were 20, 13, 17 respectively. As a matcher (Type +3), the machine would handle 13 x 3 parts and pass all 39 to the output. As an assembler (Type -3), however, it would again handle 13 x 3, but only pass 13 to the output.

Note carefully that any model utilizing a Type -3 machine (or a Type 7 to be discussed later) is essentially mixing apples and oranges, in that some buffers contain sub-assemblies, while others contain assemblies, which may themselves serve as sub-assemblies to something else. In these cases, to consider the contents of buffers independently of the physical counterpart of those contents is meaningless. Also, references to total parts resident on the line and total parts contained in all buffers must be interpreted carefully.

Any number of sub-lines can be assembled by any number of assemblers. The mechanism of this type of building block is as follows: Each assembler in the station will take exactly one part from each of the sub-lines and will pass just one "assembled" module down stream. If, when the time comes for an assembler to operate, there are no more sub-assemblies present at any of the sub-lines, the assembly will not take place, no parts are removed from any of the sub-lines, and no part is passed downstream. This is all accomplished through the following coding considerations:

a. the machine type is specified as -3,

b; the number of sub-lines acted upon is placed in the AUX field,

c. the number of the left-most buffer point acted upon is entered for the IN; the remaining input points must be numbered consecutively; the program uses the starting number and the number of sub-lines to determine which buffers to actually draw from;

d. there is still only one output point possible from the machine

e. the rate for the assembler is entered as the total number of sub-modules handled, not the number of finished assemblies handled.

A Type of +3 component still takes parts from sub-lines, but does not produce a different assembly. Rather, it simply passes all parts downstream. The sub-lines, then, can represent the same or different parts, since they all wind up the same place as distinct units. It would appear on the surface that there is no difference between this operation and simply having all the sub-lines dump into the same buffer, except that with a +3 operator, parts will be passed to the output buffer if and only if there was a part present on each sub-line. Thus, material is passed in groups of two, or three, or whatever, or it is not passed at all. This is not the same as simply letting all sub-lines enter a buffer without regard to order.

Type ±3 machine can also be used for inspection purposes, but they cannot directly be used for delay.

2.6 Type ±4: Continuous Batch Operations

An important distinction must now be made between various operational concepts. The machines discussed so far have been basically of the high-speed small-piece manufacturing type where it is not uncommon to have speeds of 40 parts-per-minute or higher. This process is continuous in the sense that the flow of finished parts never stops as long as the machine is up and running. Given a continuous input, the parts will be output (normally) one at a time every so many seconds (e.g. 40 PPM=1 part every 1.5 seconds).

There are many processes, however, which don't work this way at all. The most obvious variation being a line which produces large and expensive items at slow speeds, such as televisions, refrigerators, or cars. Here the speeds are of the magnitude of one part every 15 minutes rather than 15 parts each minute. The basic time interval for simulation then is likely to be, say, 15 minutes rather than one minute as in the situations discussed previously. Yet, this sort of operation even at these speeds is also continuous, since a new TV is still finished every so many minutes with no gaps.

A slight variation of this continuous operation exists when the machine performs some function on more than one part simultaneously. This might be, for example, a drilling function in which four drill heads operate on four separate parts at the same time for, say, 15 seconds. The output, then, is 4 x 4 or 16 PPM although no single part could ever be finished in 1/16 minute. The process is still continuous, but instead of one finished part every 15 seconds there are four. The user has a choice here of representing the process by a generalized speed of 16 PPM where no loss of accuracy is anticipated or by defining a "part" as a set of four units, in which case the speed becomes 4 PPM with this new size "part". Remember that GENMOD has no way of knowing what a "part" actually represents. It could be a single unit, a pallet of four units, 1/2 gallon, 3.6 pounds, etc. The user is really the only one who has to worry about what to call a "part".

A completely different type of operation, though, exists when a fixed amount of material is put through successive functions each of which requires a different processing time and yet the material cannot be either logically or physically split into different size chunks such that one basic time interval could be used throughout the model. Such operations are usually the rule rather than the exception on chemical lines, for example, which deal in a batch of material. This batch of, say, 200 pounds, which must remain together throughout production, might be subject to the following sequence of operations: mixing for 18 minutes, then heating for 78 minutes, then cooling for 43 minutes, etc. Characterizing such a sequence by the GENMOD building blocks presented so far would be very difficult. For that reason, the Type 4 or batch operation block was developed.

Figure 9 will serve to visualize the difference between continuous and batch operations. The observed outputs are shown in terms of their relative magnitudes and timings for

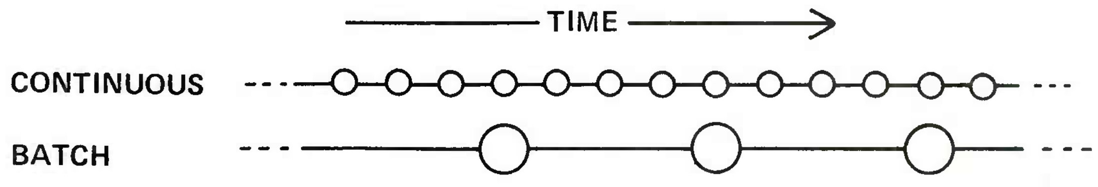


Figure 9 — Observed Output

a case whose batch size is four units requiring four basic time intervals to process. The total output is the same for both operations. The only things different are the size of the output occurrences and their frequency. Note that the batch operation itself can be thought of as continuous if the BTI and the rate are changed accordingly. For this reason, a Type 4 component acts like a continuous batch process in which small sub-batches can be thought of as being accumulated for some period of time and then the entire accumulation is released at once. In fact, this component actually removes those sub-batches one at a time from the input buffer for the specified duration, rather than remove a full batch once. The accumulated output, though, is always deposited in one chunk.

This is an important feature of this component. It is time oriented rather than size oriented. It deposits whatever batch has been accumulated after the appropriate passage of time, rather than holding everything until the appropriate batch size has been attained, an event that might never occur.

In order to code a Type 4 component it is first necessary to determine the total batch size and the process duration in terms of BTI. Although it is permissible to deal with fractional units of material during an interval, it is not possible for the user to deal with a fractional time interval. Thus, where a batch operation or any other time-dependent function is involved, the BTI must be chosen such that these timing considerations can be expressed as close to an integral multiple of BTI as possible. This may require an approximation in some instances.

Specific examples will be worked out in detail in further chapters. But, for the sake of illustration, consider the timing sequence given above: 200 pounds in three successive steps of 18, 78, and 43 minutes duration. Several choices are possible for BTI. Obviously, one minute will work, with no loss of accuracy. But, by lengthening (rather than shortening) a couple of operations slightly, other choices could work as well, as shown in the following table.

	Duration of Operation (In Intervals)		
BTI	Operation 1	Operation 2	Operation 3
1.0	18	78	43
2.0	9	39	22
3.0	6	26	15
4.5	4	18	10
6.0	3	13	8

Table 3

Once the proper batch size and BTI have been determined, the batch must be split up as if it were being processed in small chunks instead of all at once. In other words, a batch size of 120 pounds requiring 4 intervals is coded as handling 30 pounds per interval. This is necessary to simulate this sort of continuous batch operation. Note that this component actually removes sub-batches from the input buffer during each BTI, although nothing is released to the output buffer until the entire process time has elapsed. This is not the same thing as removing 120 pounds at one time, holding it for four intervals and then releasing 120 pounds. That sort of operation can be simulated in a couple of different ways which will be discussed later.

What is to happen when a Type 4 block goes down? Depending upon the intended use of the block, the user has two options:

-Type +4 will cause all material in process in that machine to be held until the machine is fixed, when processing will resume where it left off as if there were no break

-Type -4 will cause all material to be released immediately to the output buffer and a new cycle will begin from scratch when the machine comes back up.

Be aware that a Type -4 is likely to release less than a full batch whenever it goes down. Furthermore, both Types +4 and -4 are time-dependent, not batch-dependent. After passage of the process time, the accumulated batch is released regardless of whether there is a full batch size or not. As long as the input is continuous, then the resultant batch will be of the proper size. However, results vary if the input is interrupted, just as would happen in real life.

The AUX field of the data card for these machines contains the number of intervals required to complete the operation. Type ±4 machines, as such, cannot use the delay option, but they can be used for inspection and rejection.

2.7 The Delay Function

Having introduced the notion of operations that cover more than one BTI, the sequence of building block descriptions is interrupted here to discuss a feature available for several component types in GENMOD, namely a delay capability. This option allows the user to operate a component in a continuous mode, i.e. both input and output during each interval, but to delay the availability of that output downstream for a fixed number of intervals. Figure 10 shows the difference between the normal mode of operation and delay mode. In normal mode, the output is transferred to the output buffer at the end of the interval in which it was produced. In delay mode, however, the output does not arrive until a stated number of intervals later. Neither the order nor the magnitude of the outputs is disturbed, as shown by the X's which represent intervals of no production.

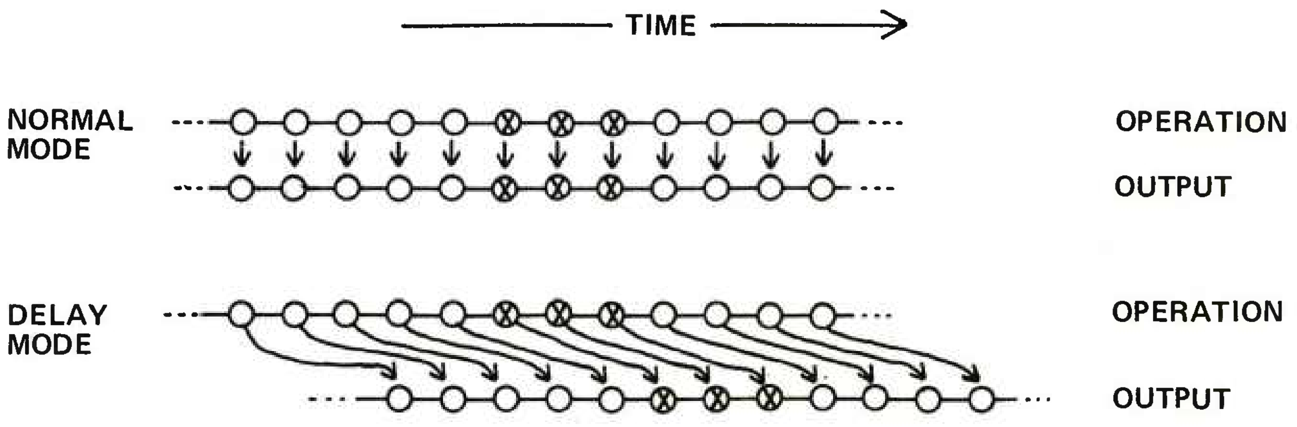


Figure 10 – Normal vs Delay Mode

There are several possible applications for the delay option. The most important of these is to represent travel time along a conveyor or short periods of storage or stabilization, for example. It can also simulate delays caused by off-line functions, such as manual operations, film processing, etc. The only thing to remember about such use, however, is that the process as described is not probabilistic. In other words, the length of the delay is constant and does not depend upon some external function.

The program handles the delay by setting up a vector of length equal to the number of intervals in question. As each interval ends, the contents of the vector are shifted forward and a new value of output is entered at the tail. The vector is considered part of the machine itself. As long as the machine is up, the values will continue shifting, to include entering null values if no parts are processed in a particular interval. Also, if parts cannot be transferred off the end of the vector, the machine will be slowed down just as if it were passing parts directly to the output buffer. If the machine is truly in down status, the vector will be likewise, i.e. the conveyor does not keep moving. In those intervals when the machine is going down or coming up, the vector will move only while the machine itself is actually up.

As the program is currently set up, up to 16 machines in a model can employ this feature, although that limit can easily be increased. Each of these machines can be assigned any integral delay length (in the AUX field) of up to 25 intervals.

The delay option words only with component types 0, 1, and 5.

2.8 Type 5: Contingencies

The component types described thus far operate from the point of view that there is only a single function being performed by that machine. It is further assumed that this function can be characterized by one set of parameters. In a lot of applications this is simply not adequate. Many times there is indeed only one physical function being performed, by which the parts are "processed", but with several non-physical considerations present that do not actually handle the parts yet which logically must be satisfied in order for the entire operation to work. Often, these contingencies can be modelled by constructing a so-called "black box" that includes both the functional device and the non-functional peripherals, with some form of combined failure rate assigned to the composite. At other times, however, it may be more desirable to deal with the device and the peripherals separately. Examples of such peripherals might include: material handlers (conveyors, positioning devices, locking pieces), feeders, control systems, calibration equipment, inspection equipment, special power sources, availability of non-continuous raw material, lubrication or clean-up equipment, etc. The general criteria for determining what could be considered as a peripheral are these:

- the function does not itself perform any work on the part
- it must be present logically in order for the process to continue
- it cannot somehow be combined with the actual operational device

A distinct component type exists to provide for an operational device attended by any number of non-operational peripherals. Note that "operational" means in terms of physically contributing to the finished product. Type 5 components work as follows: when a string of Type 5's is encountered, the last machine in the string is assumed to be the operational element with all others being the logical peripherals; if in any interval all elements in the group are up, then parts will be transferred to the output buffer of the operational element; if any machine in the group is down, no work will be performed.

These components, called grouped components, can also be used to simulate machines with dual failure or repair rates. If a machine has two or more failure modes with such widely differing distributions that there would be too great a loss of accuracy when these distributions are combined into one, then the machine can be defined as a Type 5 element with a separate peripheral for each failure mode. Since each peripheral fails independently and since the entire group will operate only if all peripherals are up and running, this arrangement will effectively sample from different failure rates for one and the same machine. Furthermore, each failure mode will have its own repair distribution. In this application, it does not matter logically which peripheral is considered the operational device, except that the last element in the group is automatically considered as such.

An example is in order. Figure 11 shows a suggested manner of diagramming Type 5 machines.

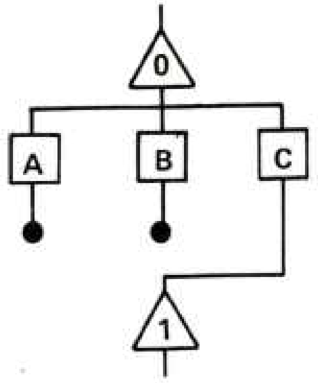


Figure 11

Machines A and B are the contingencies and do no real work on the product. Thus, their "outputs" are shown as going nowhere in particular. Only Machine C actually performs a function, and thus, only it is connected to the true output buffer.

Since all three machines are on the same level, the working of this group takes place in a single interval, which is a distinct advantage over trying to model these machines as a series string.

Since the contingencies do no work, it is not necessary to assign a speed to them. Setting the PPM field to zero on the data card is perfectly acceptable. The same is true even if the peripheral is being used to model one of several failure modes of a real working machine. Only the operational element of a Type 5 block need have a speed assigned to it.

Several specific rules must be followed when coding grouped components:

- all the elements in a particular logical group must be numbered consecutively
- all the elements in the group are designated as Type +5 except the last element, which is Type -5
- all elements in the group must have the same input point
- all elements must be assigned an output buffer; however, only the operational device is assigned a true buffer point; the others are assigned dummy buffer points, i.e. some fictitious number (e.g. 999) which is higher than the highest numbered valid buffer point; all such dummies may be assigned the same number.

The examples later in the text will illustrate the proper usage of these rules.

There is no limit to the number of peripherals that may be present in any one group. But there may at the moment only be 36 groups in any model.

Type 5 machines can use both the inspection and delay options.

2.9 Type 6: Multiple Occurrence Machines

Recall that the discussion on Type 5 machines stressed their contingency aspect, i.e. the necessity for their being logically present even if they did not actually do any work. One of the examples given was that of a control system, whose functioning is vital to the performance of the operational element. That example was presented from the viewpoint of associating one machine with one control system, namely its own. But not all machines are run by distinct control systems. More often than not, there would be one control system running several machines, i.e. the same inactive device serves as a contingency to several operational elements. How could this be modelled with the blocks described thus far?

No block discussed up to this point is capable of appearing at more than one place on a line. The Type 6 component is designed to do just this. It serves as an inactive element that can appear any number of places in the model, mainly in the role of an on/off switch that can be used to control some other operational element. This type component's up/down status is determined in the usual manner and is tested as necessary in conjunction with other machines. Thus, it must have real failure and repair distributions. When it has failed, that failure affects all references to that machine simultaneously. It is therefore best to think of Type 6 elements as single machines (perhaps off line) whose status has an influence on several other machines. Such influences are logically series operations, but are time-wise parallel, i.e. they usually do not take up a separate time step to perform.

Because it is literally impossible to have any machine physically present at two or more places at the same time, it is meaningless to think of Type 6 machines as doing real work on the product. Therefore, Type 6 machines can exist only in peripherals in a Type 5 contingency block. As such, it is immaterial what speeds are associated with them, since no parts whatsoever are actually handled and no such statistics are gathered.

Several examples will clarify the use of these Type 6 machines. Consider Figures 12 and 13 below.

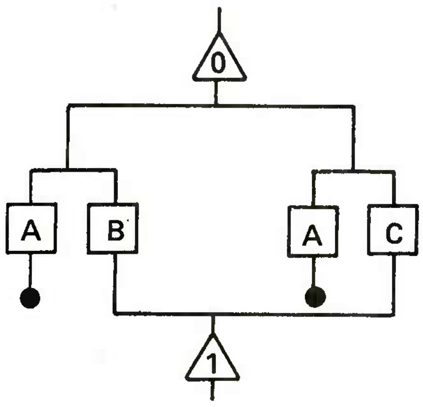


Figure 12

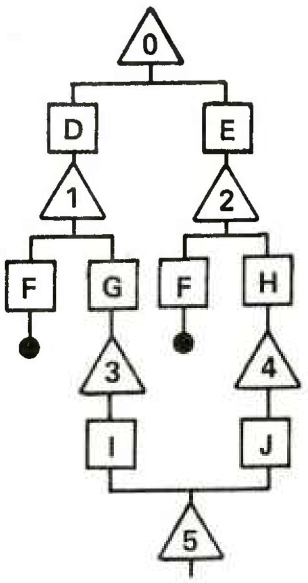


Figure 13

These models are very similar, yet show two distinct situations in which Type 6 blocks might be used. In the first case, Machines B and C work together between the same pair of buffers, provided Machine A is up. In the second, Machines G and H are also dependent upon a common contingency (Machine F), but four different buffers are involved. Except for the problems of properly assigning buffer designations and of forcing the appropriate pairs of machines to act like Type 5 groups, it is clear that Machines A and F above fulfill the requirements of a Type 6 component. Specifically, they appear more than once on the line, do no real work of their own, and their functioning affects that of other operational machines.

Recall that one of the first statements made about GENMOD machines was that in general they receive parts from one and only one buffer. This convention is violated by Type ±3 machines which serve any number of input buffers and it can easily be violated by Type 6 machines (Figure 13). Since it is usually possible to arrange a model such that the inputs to a Type ±3 element can be numbered consecutively, that technique is used with those machines. However, it is very easy to construct models where the inputs to a Type 6 machine connot be so numbered, especially in the case of a wide-ranging control system (Figure 14). Therefore, a completely different approach is needed for these machines, i.e. multiple data cards. A separate machine card must be placed into the data deck for <u>each</u> occurrence of a Type 6 machine. Each such card will have the appropriate input buffer number, a dummy output buffer number, and the AUX field will contain the number of the <u>operational</u> element with which it is associated. The <u>first</u> occurrence of the machine is signaled by placing <u>+6</u> in the TYPE field and this card must contain the true failure and repair distributions. All subsequent occurrences of the machine have <u>-6</u> in the TYPE field and the failure and repair distribution fields are ignored. For all occurrences, the DEF field is ignored, as no defective parts can possibly be produced. Remember that these machines <u>must be associated with a Type 5 machine.</u>

Consider the model in Figure 14 for specific coding information.

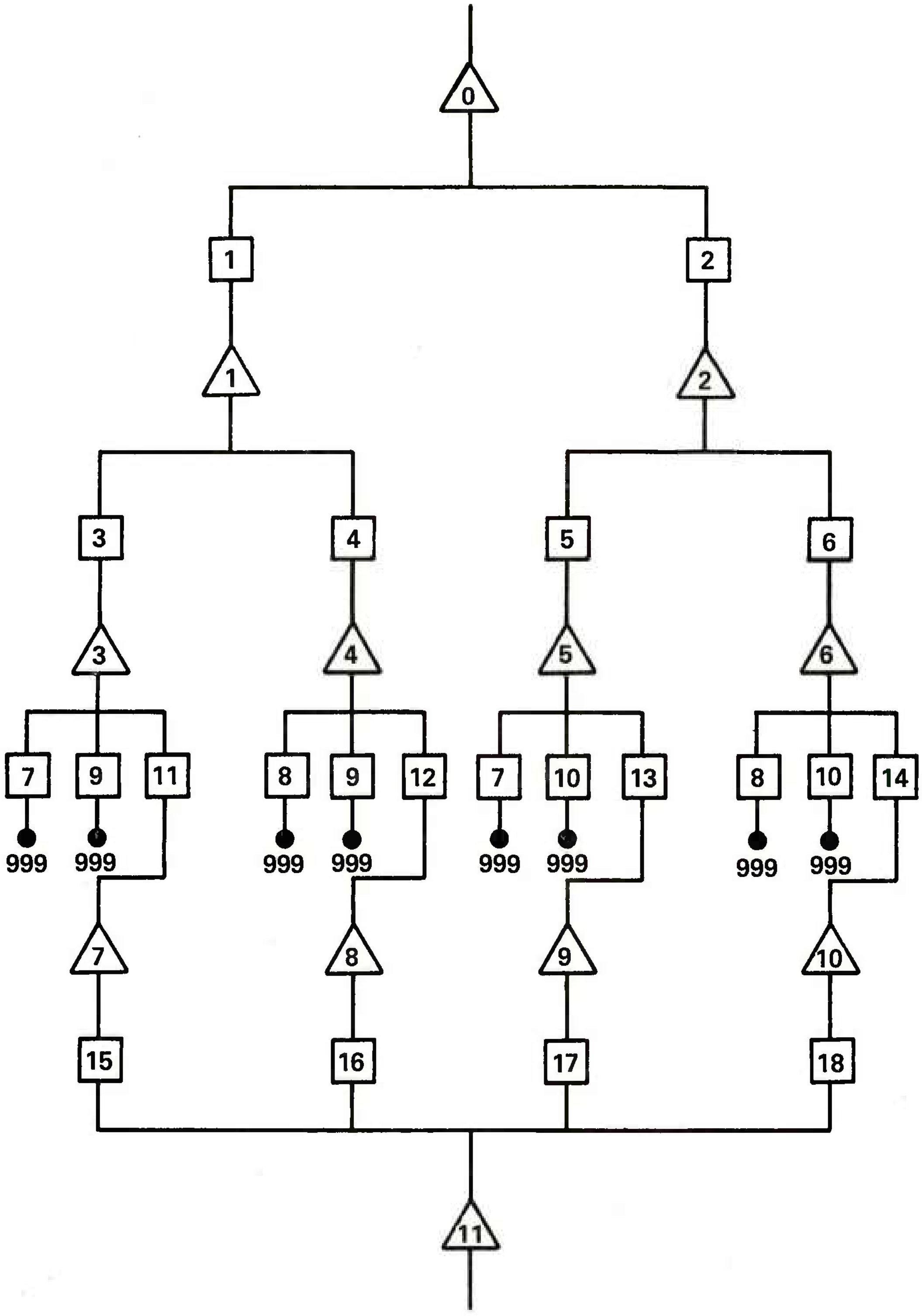

0
1
2
1
2
3
4
5
6
3
4
5
6
7
9
11
8
9
12
7
10
13
8
10
14
999
999
999
999
999
999
999
999
7
8
9
10
15
16
17
18
11

Figure 14

Here, Machines 7, 8, 9, and 10 are intended to be Type 6's, be they control systems, power supplies, feed systems, or whatever. Note that there is no way to arrange this model so that all the inputs to all the machines are numbered consecutively. There are a total of 18 machines in the model and this is the value that is punched on the appropriate data card. However, 22 machine cards are needed to define this line. Using purely hypothetical rates and distribution data, the following table shows how the cards for Machines 7 thru 14 should be punched (not all fields are shown, however).

MACH	IN	OUT	PPM	MTBF	MTTR	DEF	TYPE	AUX
7	3	999	0.0	60.0	5.0	0.000	6	11
9	3	999	0.0	60.0	5.0	0.000	6	11
11	3	7	100.0	75.0	10.0	0.015	-5	0
8	4	999	0.0	60.0	5.0	0.000	6	12
9	4	999	0.0	0.0	0.0	0.000	-6	12
12	4	8	100.0	75.0	10.0	0.015	-5	0
7	5	999	0.0	0.0	0.0	0.000	-6	13
10	5	999	0.0	60.0	5.0	0.000	6	13
13	5	9	100.0	75.0	10.0	0.015	-5	0
8	6	999	0.0	0.0	0.0	0.000	-6	14
10	6	999	0.0	0.0	0.0	0.000	-6	14
14	6	10	100.0	75.0	10.0	0.015	-5	0

Note that the machine cards do not have to be punched in numerical order, as long as a value is entered in the MACH field. (This is true for <u>all</u> GENMOD models.) However, these particular cards must be entered in the sequence shown in order for the appropriate Type 5 groups to be generated. Also note the dummy buffers numbers, 999. This model will be referenced again later when fully coded examples are introduced.

In summary, a Type 6 machine is a multiply-occurring, inactive contingency whose performance influences successful functioning of more than one operational element. It must be made part of a Type 5 contingency group.

2.10 Type ±7: Input Converter (Packer/Unpacker)

With the exception of a Type -3, all machine types introduced so far pass to their output buffer the same number of parts that were removed from the input buffer (less defects, of course where applicable). Only Type -3 was capable of taking X apples and calling them Y oranges, provided at least Y apples existed at each of the X/Y input buffers. But this process was one of matching, part for part, in an assembly operation. It was not intended solely to reduce the number of parts from X to Y. Furthermore, it operated on more than one input buffer. No type component yet described works on a single buffer and reduces or expands the volume of parts passing through it. This new block, Type 7, is designed to do so.

The main purpose for a Type 7 machine is to simulate a packing or unpacking operation, in which the output at each interval represents a fixed fraction or multiple of the input. To work properly, the process must be describable as relabelling so many identical items as a single new part, with all such items coming from one input source (i.e. no matching is involved), or of expanding a single item into several smaller ones.

Examples of such a packing or reducing operation might be eggs in a carton, wheels on a car, layers on a cake, boxes on a pallet, batteries in a flashlight, and so on. Oranges in a crate is not, however, a packing in this sense, since there is not usually a fixed fraction involved. Also, integrated circuits onto a breadboard is not, because usually one each of several different circuits is involved (i.e. matching). If one is willing to generalize by using an average fraction, then oranges in a crate can be packed with a type 7 machine. Likewise, if keeping track of specific circuits is not crucial, then this machine can also put a breadboard together.

Examples of unpacking or expanding might include chopping a big log into little logs, cutting a roll of steel into sheets, unloading cartons of sub-assemblies, stripping like sub-components off scrap assemblies, etc.

In all uses of Type 7 components, so many units are converted into something else by a fixed ratio. This integral number of units is placed in the AUX field of the data cards. The total number of units handled per interval is still determined, though, by the speed of the machine. Thus, it is possible to wind up processing a fractional number of containers per interval, but never a fractional number of units per container.

For all practical purposes, these machines work exactly like Type 0 and 1 except that, just before the parts are transferred, they are divided or multiplied by whatever value is specified in the AUX field. In this regard, if two or more Type 7 machines are placed in a station, they compete for the inputs, just like Type 0. Type +7 will accomplish a reduction in input (packer) while a Type -7 will accomplish an expansion in input (unpacker).

For the sake of illustration, suppose 350 pieces are originally in an input buffer that is worked upon by a Type 7 machine rated at 120 PPM, packing them 24 to a box. If no more pieces enter the input buffer and the machine does not go down, the following table shows the status of both the input and output buffers for the first few intervals of operation.

Interval	Parts Handled	Input Buffer	Parts Passed	Output Buffer
0	—	350.00	—	0.00
1	120.00	230.00	5.00	5.00
2	120.00	110.00	5.00	10.00
3	110.00	0.00	4.58	14.58
4	0.00	0.00	0.00	14.58

Just as with Type 3 components, the speed for converters is given in terms of original (input) parts handled, rather than final (output) units. Also, since the term "part" is being applied to two different items on lines involving Type 7 machines, care must be used in interpreting buffer contents.

These converter machines can be used for the rejection of defective material, but cannot use the delay option.

2.11 Type 99: Non-Existent Machine

One of the more useful benefits available from a simulation is the ability to study the effect of adding or removing machines at various places in a line. For example, having decided to place five machines in a particular station, it is certainly worthwhile knowing whether four could do the job as well, or that six would provide a better margin of error.

To develop a separate GENMOD model for each of these variations is not necessarily the best approach, in as much as extensive renumbering of machines and buffers is usually involved. An option is available to the user, however, that handles such modifications provided they are anticipated beforehand. Any machine can be effectively removed from a GENMOD run simply by changing its type code to 99.

The user can take advantage of this option very easily by placing in the initial model the maximum number of machines considered feasible. Then on a particular run, turn off one or more of these machines as needed. Type 99 machines are completely ignored on a run. They are not listed, they do no work and no statistics are gathered about them.

As easy as this option is to use, caution must be exercised in certain situations, as shown in the following figure.

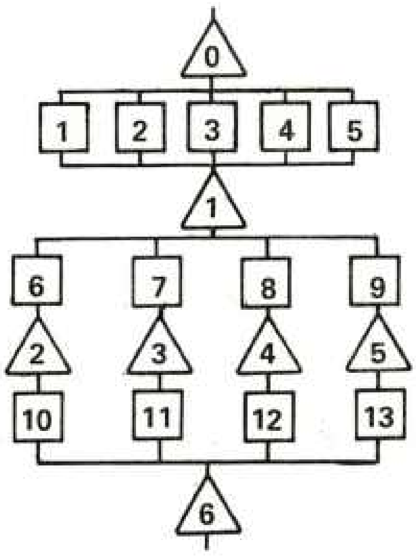


Figure 15

Leaving all these machines active will allow study of the full model. Turning Machine 5 off will show the effect of only having four machines in the first station. In the second station, however, we must be careful to turn off an entire branch. Just turning Machine 9 or 13 off without its mate would cause erroneous results. Note that turning off the 9/13 branch above will effectively make Buffer 5 inactive. No special action is required for this eventuality, though, as the statistics will automatically reflect the fact that that buffer is never used.

Similar care must be used when dealing with machines that are logically associated with others (e.g. Types 2, 3, 5 and 6). Turning off one or more of these machines might require reassignment of certain parameters for the remaining machines.

Whenever a machine is turned off by means of Type 99, it still must be accounted for in the input deck. A card must be present for it and it must be included in the count of machines on the line. Since all variables other than the type code are ignored for these machines, it does not matter what is punched there. So it is perfectly acceptable to leave all remaining data on the card as is.

2.12 Dummy Machines

Refer again for a moment to Figure 15. Suppose that originally Machines 6 through 9 were intended to be some sort of a setup operation, in which parts are aligned properly before passing to Machines 10 through 13. Suppose further that this latter station of machines is then redesigned to include this setup operation, thereby making the preceding step unnecessary. Without resorting to actually redrawing the model, how can this first bank of machines be bypassed? Designating them as Type 99 is not the answer, since then nothing would ever pass out of Buffer 1 to the succeeding machines. What is needed is a method of making the unnecessary machines act as perfect transferrers of parts, never going down and always moving whatever material is expected of them. In other words, create dummy machines that always do what is required.

Such a dummy machine is established in GENMOD simply by setting the MTTR equal to 0.0. Coding within the program will then prevent that machine from ever going down. Being up then, in every interval, the machine will transfer parts at a rate determined by its usage.

With one exception, to be discussed shortly, dummy machines will act like Type 0 or 1 components, depending on how many are present in a station. A single machine will act like Type 1, in that it will transfer the minimum of its rated speed, the contents of the input buffer, and the available space in the output buffer. When two or more are present, they will act like Type 0 and share the input equally before transfer.

Because dummy machines will always transfer as many parts as are available and there is room for, up to its rated speed, this speed should be assigned carefully. It is usually best to set the speed the same as the machine or station that follows.

Taking the place of a previously present machine should not be construed as being the prime purpose of a dummy machine, nor should indiscriminate use of such dummies proceed unchecked. The reason for these reservations is that dummy machines take a full time interval to operate and in general insert an additional buffer point into the model. The main use for a dummy machine is to space out the timing of events in those models where timing is important. Since a dummy machine essentially does nothing more than move a quantity of parts and take one time interval to do so, each occurrence of a dummy simulates the passage of time of one interval. Putting several dummies back to back can be used to represent the passage of any integral multiple of BTI.

Such passages of time might be used to reflect travel from one building to another while some other operation is taking place elsewhere, or it might be used to balance out the lengths of two parallel branches with a different number of inherent time steps in each. To be sure, such detailed concern over timing aspects might seem premature, but the reader should be aware that a GENMOD model can be made as simple or complex as the user's requirements dictate. Illustrations of timing problems will be covered when some actual models are constructed.

Because of the simplistic nature of dummy machines, they cannot use the defect or delay options. They are merely untiring machines that transfer parts perfectly.

2.13 Proportional Splitters

The discussion on dummy machines above mentioned that there is one exception to their otherwise straightforward operation. When two or more are placed together in a station, they will automatically split the input evenly among themselves and transfer an equal share, given that their speeds are the same and there is sufficient room in the output buffers. There may be occasions, however, when distributing these shares evenly is not desired. Instead of each of four machines transferring 25%, for example, it may be desired to move 20%, 40%, 30%, and 30% respectively in that order. This cannot be accomplished simply by adjusting the speeds or by making the dummies Type 1 instead of Type 0. Juggling speeds and using Type 1 designations will fail to produce the desired effect if the input buffer ever falls below the sum of the speeds of the machines involved. A completely different type of operation is needed here: a proportional splitter.

A feature has been built into GENMOD whereby the contents of an input buffer can be distributed to two or more output buffers according to a fixed proportion. This is done by inserting a set of dummy machines with the specific proportion each is to handle placed in the DEF field of the card. Recall that dummy machines cannot reject material, so that the values in the DEF field are not considered defect rates, but rather proportions. This will only happen, though, with dummy machines, i.e. ones with MTTR = 0.0. Furthermore, the type code should be designated as Type 0.

Once again referring to Figure 15, Machines 6 through 9 can be used to deliberately split the material from Buffer 1 to Buffers 2 through 5 in whatever ratios are desired, such as 20/40/30/30.

This proportional splitting is not deterministic, i.e. the same proportions will be used every interval. The amount of material actually transferred, however, may not necessarily be exactly this amount. Two situations can cause a variation:

a. If the amount scheduled for transfer is greater than the rated speed of a machine, or

b. If there is insufficient room available in the output buffer.

In both cases, the material transferred will be cut back for the machine affected. But, the proportions already allocated to the other machines will not be readjusted.

While the primary use of proportional splitters is to distribute material properly to branches of varying capacity, another valuable use is to simulate feedback loops by which a certain percentage of the output is sent back upstream. Such loops are useful for recycling material in chemical lines, but can also represent "slightly" defective units being sent back for rework. Examples employing feedback loops will be presented later.

No internal check is made by GENMOD to see if the designated proportions add up to 100%. The user must assure that the speeds of the machines are adequate to handle whatever share is directed toward them. Finally, because of the occasional fluctuations, explained above, no attempt is made to assure the correct proportions are maintained. The amounts allocated for distribution are based upon the initial contents of the input buffer at the start of the interval.

Type Code	Components	No. of Inputs	No. of Outputs	Defect Capability	Delay Capability	Speed	Use of AUX Field
0	Cooperative	1	1	Yes	Yes	Actual	Delay Time
1	Competitive	1	1	Yes	Yes	Actual	Delay Time
±2	Sequential	1	1	Yes	No	Actual	Cycle Duration
±3	Assembler	N	1	Yes	No	N x Actual	No. of Inputs
±4	Batch	1	1	Yes	No	Actual/N	Cycle Duration
+5	Grouped-Peripheral	1	NA	NA	NA	NA	NA
-5	Grouped-Operational	1	1	Yes	Yes	Actual	Delay Time
±6	Multi-Occurrence	N	NA	NA	NA	NA	Related Machine
±7	Converter	1	1	Yes	No	N x Actual	Conversion Factor
99	NO OP	NA	NA	NA	NA	NA	NA
(0,1)	Dummy	1	1	No	No	Actual	NA
(0,1)	Proportional Splitter	1	1	No	No	Actual	NA

TABLE 4 — Summary of Block Characteristics

This page intentionally left blank.

3. BUILDING MODELS

3.1 Rules of the Game

In this section we shall begin to construct actual GENMOD models, preparing complete decks of input cards, running the models, and examining the results. These models will start out very simply and then increase in difficulty as more features are added. Whereever possible, the same basic line shall be followed through numerous modifications to show the effect of incorporating different options available.

Throughout these examples, however, certain guidelines shall be adhered to. Only after the basic techniques of GENMOD have been explored thoroughly will these rules be deviated from in an attempt to demonstrate what is or is not sacred. These guidelines include the following.

a. All machines will be numbered from top to bottom, left to right, beginning at 1. No numbers will be deliberately skipped except to show the use of Type 99 machines.

b. Buffers will be numbered from top to bottom, left to right, beginning at 0. The highest numbered (real) buffer will always be the final output point.

c. Initially, a basic time interval of one minute will be used.

d. Timing considerations and spacing of machines will be ignored for the moment, although the consequences of selecting a particular configuration will be pointed out.

e. Rather than tie the examples to specific lines or machines, the category titles used will generally be nondescript terms such as OPER1 or MACHB, and so on. Where a particular type block is being explained it will be called by such terms as DUMMYA, ASSEMBLER2, etc. The reader should be aware, though, that any title can be used.

3.2 Straight Series Operation

We shall begin with a very simple line of four sequential steps of one machine each, with no funny business. This might represent, say, four operations in the manufacture of a pencil. Huge bins of nearly finished pencil stock are presented to the first machine which trims them to their final dimension and sends them to another bin. The second machine grabs trimmed pencils from this bin and spray paints them on a continuous belt that ultimately drops them in a third bin. From this, the third machine positions them and stamps the company name on them before passing them to a fourth buffer that maintains them in proper alignment. The last machine takes each pencil and crimps on the eraser and end cap and then dumps them in a big box.

Our first job is to draw a schematic of this operation, label everything, and then consider what sort of assumptions or constraints we have built into the model. Figure 16 depicts this pencil line.

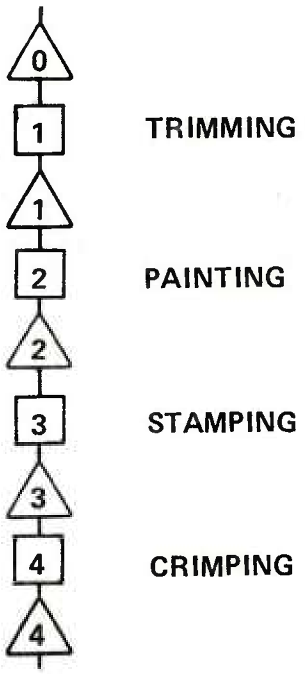


Figure 16

Even with something this simple, we find that an awful lot of generalizations have been made.

a. The raw pencil stock is fed in at Buffer 0 and we assume that there will always be enough there for the trimming machine to handle. Later examples will show how to simulate shortages in raw material.

b. The mechanisms that pick up individual pencils and redeposit them in the bins are assumed for the moment to be part of the machines themselves.

c. A one-minute interval requires that the paint dries in this time. We will live with this for the moment and adjust it in later modifications.

d. Buffer 3 must hold the pencils in proper alignment rather than in random orientation. We will assume this is done automatically.

e. Where are the paint and erasers coming from? From now we will say that there will always be paint and an end cap with eraser available. Later we will tackle this problem by means of contingencies.

f. Are the pencils really just dumped in a box at the end? Probably not, but once they leave Machine 4 they are no longer of any interest to <u>this</u> model. We'll add further machines later.

g. As set up now, it will take four minutes for a raw pencil to get dumped out at the end of the line. Is this realistic? For now we will say yes, mostly because the volume of pencils produced is so great as to override any inaccuracy that might be introduced by this constraint. As detailed a discussion as possible will be presented shortly on <u>exact</u> timing considerations.

Having drawn the diagram, it is clear that we have defined four categories of machines, with but a single machine in each. Also, there are only four buffers to worry about, as the raw material point (Buffer 0) will take care of itself.

These four machines shall all be made Type 1, with no delay and no defect rate. They will be given exponential failure and repair distributions, with approximately the same availabilities, and the rated speed shall be in keeping with an expected output of 54 PPM. All this is reflected in the following table.

MACH	IN	OUT	PPM	MTBF	MTTR	DEF	TYPE	AUX
1	0	1	82.50	36.0	4.0	0.0	1	0
2	1	2	74.25	48.0	6.0	0.0	1	0
3	2	3	66.00	40.0	4.0	0.0	1	0
4	3	4	60.00	45.0	5.0	0.0	1	0

One approximation for the expected buffer sizes can be calculated from the following relationship:

Expected Size = (MTTR of exiting machine) X (Speed of entering machine) X (Availability of entering machine)

We will inflate these by a safety margin of 10% and seed them initially with no parts. Our buffer table then looks as follows.

BUFFER	SEED	MAXIMUM
1	0.0	490.0
2	0.0	290.0
3	0.0	330.0
4	0.0	∞

Note that Buffer 4, the output point, has an infinite maximum. This is indicated, actually, by placing a 0.0 on the data card. Also, an output buffer should never be seeded as this only distorts the simulated output.

Since the first machine can't work faster than 82.5 PPM, this is all that will be fed it in terms of raw material.

We will say that this line works continuously on a 420-minute shift. But, for the sake of brevity, we will only run this model now for 30 minutes. For each of these intervals, though, we will print out complete machine handling and detailed buffer status, things that are not normally done with such frequency.

A title for this line is easy to assign and the category titles are self-explanatory.

Putting all this together, then, the following is a listing of all the data cards needed for this model.

```
0000000000000000000000
ACME PENCIL FINISHING LINE
004
TRIM      PAINT      STAMP      CRIMP
004
  1  1  1  0  1   82.50   36.00   0.000   4.000   0.000   0.000  1  0
  2  2  1  1  2   74.25   48.00   0.000   6.000   0.000   0.000  1  0
  3  3  1  2  3   66.00   40.00   0.000   4.000   0.000   0.000  1  0
  4  4  1  3  4   60.00   45.00   0.000   5.000   0.000   0.000  1  0
004
  1   0.00000   490.000    1
  2   0.00000   290.000    1
  3   0.00000   330.000    1
  4   0.00000   0.00000    1
   82.5000   420.000    0    0    0    0   1.00000
0000000000000000000000
   30   1     1    2    0
0000000000000000000000
```

Figure 17 — Example 3.2 Input Deck

(The three lines of 20 zero's represent End-of-Record/ End-of-File markers as printed by CDC-6600 system.)

This input deck, as well as all GENMOD decks, is divided into two logical records. The first, containing in this case 14 cards, resides on logical file TAPE1 and represents the description of the actual model itself. The second, always consisting of but a single card, resides on logical file TAPE5 and contains the information needed to execute the model for a given number of intervals, using a given set of print options.

This division into two distinct files permits storage of the model on a permanent file device. At execute time this model, usually consisting of many more lines than in this example, is retrieved from memory rather than from punched cards. It is then simply matched up with the applicable execute card. In this way, the same model can be run many times under varying conditions without resorting to reading a bulky card deck more than once.

Before proceeding with the execution of this model, let us first examine thoroughly the makeup of these two logical files, using the above deck for reference.

TAPE1 always contains five sections of information:

a. TITLE

- one single card containing some descriptive reference, e.g. "ACME PENCIL FINISHING LINE"

b. CATEGORIES

- one card containing a 3-digit integer representing the number of category titles used in the model, e.g. "004"

- as many cards as are needed to list these category titles, 8 to a card, 10 columns each, e.g. "TRIM   PAINT   STAMP   CRIMP"

c. MACHINES

- one card containing a 3-digit integer representing the number of machines in the model, including dummies and no op's, e.g. "004"

- one card for each such machine, in the format described in para 2.1

- N.B. even though more than one card is needed for each Type 6 machine, the machine itself is included only once in the count of machines in the model

d. BUFFERS

- one card containing a 3-digit integer representing the number of buffers in the model, not counting Buffer 0; this number is usually the same as the number of the output buffer; e.g. "004"

- one card for each buffer, containing the buffer number, the initial seed, the maximum capacity, and a print code, in the following format:

Number	Seed	Maximum	(blank)	Code
I3	F10.0	F10.0	4X	I1

Code = 0 or blank = Don't Print Buffer Status
Code = 1 = Do Print Buffer Status (Up to 15 Buffers)

e. <u>OPERATING ENVIRONMENT</u>

— one single card containing seven parameters relative to proposed operation of the line

(1)	RAWMAT	— amount of raw material available at head of line every interval	
(2)	SHIFTL	— scheduled shift length, in terms of BTI	
(3)	IMODE	— mode of line operation	
(4)	IDUR	— frequency of maintenance action	See Table 5 for values
(5)	IPURGE	— code for purge policy	
(6)	ICLEAR	— code for clear policy	
(7)	BTI	— real time equivalent for each simulation interval	

These values are punched in the following format:

RAWMAT	SHIFTL	IMODE	IDUR	IPURGE	ICLEAR	BTI
F10.0	F10.0	I5	I5	I5	I5	F10.0

This page intentionally left blank.

TABLE 5 – SELECTED INPUT CODES

IMODE	=	0	=	Continuous Operation (Default) (i.e. no maintenance performed)
	=	1	=	Repair all machines every IDUR intervals (i.e. corrective maintenance at fixed intervals)
	=	2	=	Repair all machines and reset times-to-next-failure every IDUR intervals (i.e. preventive maintenance at fixed intervals)
IPURGE	=	0	=	Leave buffers alone when a failure occurs (Default)
	=	1	=	Purge contents from all buffers when any failure occurs
ICLEAR	=	0	=	Leave buffers alone at end of each shift (Default)
	=	1	=	Clear buffers every IDUR intervals
NSPEC	=	0	=	Full output - all tables printed (Default)
	=	1	=	Interval-by-interval machine production also printed
	=	2	=	Summary tables only printed
IREW2	=	0	=	Buffer summary printed (Default)
	=	1	=	No buffer summary printed
	=	2	=	Reduced buffer summary (no quartiles printed)
IREW3	=	0	=	Logbook printed (Default)
	=	1	=	No logbook printed

This page intentionally left blank.

TAPE5 always contains a single card, the EXECUTE card, with five pieces of information:

a. NSIMUL	–	length of simulation, in intervals	
b. NFREQ	–	frequency of buffer and/or machine handling status prints	
c. NSPEC	–	code for specific printouts desired	See Table 5 for values
d. IREW2	–	code for buffer status print, on TAPE2	See Table 5 for values
e. IREW3	–	code for logbook print, on TAPE3	See Table 5 for values

These values are punched in the following format:

NSIMUL	NFREQ	NSPEC	IREW2	IREW3
I5	I5	I5	I5	I5

We can now start to examine a typical run of this model for the indicated 30 intervals. Be aware that this printout is unique and cannot be repeated, even with the exact same input deck. The very nature of a simulation calls upon generating sets of random numbers. The standard version of GENMOD automatically advances to a <u>different</u> starting point in the generating function. Only by complete accident could two GENMOD runs ever be the same. A version of GENMOD that repeats random number sets or employs user-specified seeds is available upon request if needed.

The first page of a standard GENMOD output (NSPEC≠2) contains essentially an echo of the input data along with the calculated times-to-first-failure for all machines. It is divided into four basic sections.

a. Title and printout of all machine parameters, including availability and sequencing within categories. If any Type 2 or Type 5 elements are used in the model, an additional paragraph of information appears in this section.

b. Times of first failure for all machines, as sampled from the specified failure distribution. This is essentially the first calculation done by GENMOD. Dummy and Type 99 machines will show up with infinite times to first failure, i.e. ******. This string of stars will show up for any real machine whose time to first failure is beyond 1,000,000 intervals. Such machines, of course, can never be observed to have failed.

```
        ACME PENCIL FINISHING LINE

                                INPUT   OUTPUT             FAILURE          REPAIR        NOMINAL   COMP    AUX    DEFECT
MACHINE  OPERATION(SEQ)        BUFFER   BUFFER    RATE    MEAN   SIGMA    MEAN   SIGMA    AVAIL.    TYPE   FIELD    RATE
-------  --------------        ------   ------    ----    ----   -----    ----   -----   -------    ----   -----   ------
   1     TRIM         ( 1)        0        1      82.5    36.0     0.0     4.0     0.0     .9000      1       0    0.00000
   2     PAINT        ( 1)        1        2      74.3    48.0     0.0     6.0     0.0     .8889      1       0    0.00000
   3     STAMP        ( 1)        2        3      66.0    40.0     0.0     4.0     0.0     .9091      1       0    0.00000
   4     CRIMP        ( 1)        3        4      60.0    45.0     0.0     5.0     0.0     .9000      1       0    0.00000

TIMES OF FIRST FAILURE
----------------------
  (  1=     11)   (  2=     36)   (  3=     41)   (  4=     43)

CONTROL POINT DATA
------------------

CONTROL   INITIAL    MAXIMUM    STATUS                    MAXIMUM  MAXIMUM
 POINT    CONTENTS   CAPACITY   PRINT    ENTRIES  EXITS    INPUT    OUTPUT
-------  ---------   --------   ------   -------  -----   -------  -------
   1         0.         490.     YES        1       1       82.5     74.3
   2         0.         290.     YES        1       1       74.3     66.0
   3         0.         330.     YES        1       1       66.0     60.0
   4         0.      ********    YES        1       0       60.0      0.0   (OUTPUT POINT)

        RAW MATERIAL AVAILABLE=        82.50 PER INTERVAL
        SHIFT LENGTH=                 420.00 INTERVALS
        LENGTH OF SIMULATION=          30    INTERVALS
        BASIC TIME INTERVAL=            1.00 MINUTES

-----------------------------------------------------------------------------------------------------------------------
```

Figure 18 — Example 3.2 — Page 1

c. Buffer or control point data, including number of machines entering and exiting and maximum apparent input and output. These values are very helpful in debugging a new model.

d. Statement of basic input parameters, including raw material, shift length, duration of simulation, and BTI.

The dashed line indicates the end of the input data. All further output is generated after the actual simulation begins.

The nature of the next section of output depends upon the parameters NSPEC and NFREQ. This printout resulted from values of one for both variables, namely print a Periodic Machine Handling Table and update it every interval. It should be obvious that printing this table for large models executed for long durations will result in voluminous output. But, this table is very useful for debugging sections of a model.

Looking at this table carefully will reveal several interesting realities of a GENMOD simulation that the users should get accustomed to:

a. Starting with an initial seed of 0 parts in all buffers, the program has no choice but to allow only Machine 1 to operate in the first interval, since it is the only machine being fed any parts (from the raw material point, Buffer 0). At the end of the first interval, these 82.5 (rounded for printout purposes to 83) parts are transferred all at once to Buffer 1 (as will be seen when the Buffer Status Table is examined).

b. In the second interval, through, both Machines 1 and 2 have parts to draw from. Therefore, both are shown as handling parts. But note that there is a difference in speed of 8.25 parts. These excess parts will be left in Buffer 1 to accumulate. In the third interval, the first three machines work and an excess of 8.25 parts will also start accumulating in Buffer 2.

c. It isn't until the fourth interval that Machine 4 gets to operate and finally deposit pencils in the output buffer. Note that even here an excess of six parts per interval will accumulate in Buffer 3.

d. From the table of times to first failure it can be seen that the earliest failure is in the 11th interval. Therefore, from the fourth through tenth intervals a sort of steady state exists, with each machine working at a constant pace and with the various buffers filling up at the rates pointed out above. This situation would continue until either some buffer becomes full or else some machine fails. It is this latter situation that occurred on this run. During this 11th interval, Machine 1 went down about halfway through, permitting processing of 44 parts. Note that GENMOD does not have machines fail or get repaired precisely at the start of the scheduled interval. That action actually takes place at some random point in that interval. The discussion later on the Logbook touches on this again.

```
              ACME PENCIL FINISHING LINE

              PERIODIC MACHINE HANDLING
              -------------------------
                    1       2       3       4
         1         83.      0.      0.      0.
         2         83.     74.      0.      0.
         3         83.     74.     66.      0.
         4         83.     74.     66.     60.
         5         83.     74.     66.     60.
         6         83.     74.     66.     60.
         7         83.     74.     66.     60.
         8         83.     74.     66.     60.
         9         83.     74.     66.     60.
        10         83.     74.     66.     60.
        11         44.     74.     66.     60.
        12         74.     74.     66.     60.
        13         82.     74.     66.     60.
        14         83.     74.     66.     60.
        15         83.     74.     66.     60.
        16         83.     74.     66.     60.
        17         83.     74.     66.     60.
        18         83.     74.     66.     60.
        19         83.     74.     66.     60.
        20         73.     74.     66.     60.
        21         18.     74.     66.     60.
        22         83.     74.     66.     60.
        23         83.     74.     66.     60.
        24         83.     74.     66.     60.
        25         83.     74.     66.     60.
        26         83.     74.     66.     60.
        27         83.     74.     66.     60.
        28         83.     74.     66.     60.
        29         83.     67.     66.     60.
        30         83.     66.     66.     60.

     TOTAL PARTS HANDLED BY EACH MACHINE
     -----------------------------------
      (  1=      2353.)  (  2=      2138.)  (  3=      1848.)  (  4=      1620.)

     TOTAL RAW MATERIAL CONSUMED=     2353.
     TOTAL MATERIAL IN-PROCESS=        733.
```

Figure 19 — Example 3.2 — Page 2

e. Machine 1 was repaired and brought back on line in the very next interval, i.e. the sampled time to repair was one interval. GENMOD will never allow a time to repair or a time to next failure be less than one interval. This is important to remember when choosing a BTI, as that influences the likelihood of a sampled failure or repair turning out less than one interval. In this example, the machine was restored to service in time to handle about 74 parts.

f. Again a steady state situation settles in until the next major action, a second failure of Machine 1 in interval 20, this time after having had time to handle 73 parts. Again the repair took only one interval and Machine 1 is brought back up in interval 21 in time to process 18 parts.

g. The next period of steady production lasted until the 29th interval when Machine 2 was forced to slow down because Buffer 2 became full. Apparently the choice of a maximum size of 290 was inadequate for <u>this</u> run. Before being tempted to change this maximum size, however, several additional runs of much longer duration should be made and the long term average considered. Machine 2 was again slowed in the 30th interval, the end of our run, and would continue to be slowed down until something else happened, which in this case would have been the actual failure of Machine 2 itself in the 36th interval.

The next item in this section of the output gives the total parts actually handled by each machine for the entire simulation run. These totals are expressed in terms of the units each machine works on. For this example everything is, of course, simply pencils. But, the situation is much more complicated on lines with Type 3 or 7 components. As expected, the totals for Type 5 (Peripheral), Type 6, Dummy, and No Op machines will show up as zero. In all cases, these totals are <u>gross</u>, i.e. before removal of any rejects, where applicable.

The final two lines in this section relate a pair of handy totals, raw material taken in at all raw material points and material still resident in all buffers other than the output buffer. The total raw material consumed should match the total parts handled by all machines at the head of the line or all independent branches. Here that is simply Machine 1. The total material in-process represents how many partially finished parts are sitting around in buffers, including material under control of a DELAY-option machine, i.e. material released by a machine but not yet transferred to any buffer. This total is dimensionless in that it mixes apples and oranges that would result from using Type 3 or 7 elements.

In a straightforward line such as this, total raw material consumed (2353) should equal total output (1620) + total material in-process (733) + total rejects (0). This equation, though, may be slightly affected by roundoff problems.

Because buffer status print is a user-specified option and is accumulated on a separate internal tape, it does not appear next on the standard output. Rather, machine utilization statistics are printed. Two sets of observed availability data are given, by machine and by category. Normally, where there are more than one machine in a category present on a line, these two tables are different. However, in this model, the information is simply duplicated. Presented are the total failures, total downtime, and total uptime for each machine. Note that total downtime and total uptime must add up to total simulation time, i.e. GENMOD expects all machines to be ready for work in all intervals even if in certain circumstances they logically cannot. What total downtime really shows is the number of intervals at the start of which each machine was in a down state. Thus, in our example, Machine 1 was down at the start of intervals 12 and 21 only. Thus, it had two intervals of downtime. No other machines failed at all during this short demonstration run. The availabilities therefore are 1.0000 for all machines except Machine 1. This availability figure is, of course, operational availability in that it is (total operational time)/(total scheduled time).

To give some idea of the efficiency of a particular model, a summary of stop times is also given for each machine. These times reflect a count of the number of intervals during which a machine was slowed or stopped for some reason other than its own failures. These stoppages are of two types only. Downstream stop time represents a full output buffer. As noted in the narrative above, Machine 2 had to slow down twice due to Buffer 2 being full. Thus, it has two intervals of downstream stop time. Upstream stop time represents an empty input buffer. Recall that the flow of parts through GENMOD is such that Machine 2 could not work until the second interval, Machine 3 in the third and so on. Therefore, Machines 2, 3, and 4 are shown as having 1, 2, and 3 intervals of upstream stoptime respectively. Also shown is total stop time, which can be deducted from total uptime to get an estimate of machine efficiency based upon the utilization imposed upon it by the model.

Bear in mind that GENMOD considers failures more important than slowdowns. Therefore, if a machine is scheduled to go down due to its failure distribution, it will do so even if its input buffer is empty and might otherwise have been left in an idle state rather than be forced to work.

Shown between these two availability tables is the total parts rejected during the run. It is, of course, zero for this example. When parts are rejected, they are done so as individuals without regard to what units they might represent, i.e. again apples and oranges are mixed.

```
AVAILABILITY DATA          (BY MACHINE )
-----------------
                            TOTAL     TOTAL      TOTAL      AVERAGE        S T O P    T I M E S
MACHINE  OPERATION(SEQ)   FAILURES  DOWN TIME   UP TIME   AVAILABILITY  DOWNSTREAM  UPSTREAM  TOTAL
-------  --------------   --------  ---------   -------   ------------  ----------  --------  -----
   1     TRIM      ( 1)      2          2         28         .9333          0          0        0
   2     PAINT     ( 1)      0          0         30        1.0000          2          1        3
   3     STAMP     ( 1)      0          0         30        1.0000          0          2        2
   4     CRIMP     ( 1)      0          0         30        1.0000          0          3        3

TOTAL UNITS REJECTED=     0.

AVAILABILITY DATA          (BY CATEGORY)
-----------------
                  NO. OF    TOTAL     TOTAL      TOTAL     AVERAGE       AVERAGE
     CATEGORY    MACHINES  FAILURES  DOWN TIME  UP TIME  AVAILABILITY   STOP TIME
     --------    --------  --------  ---------  -------  ------------   ---------
( 1) TRIM           1         2         2         28        .9333          0.0
( 2) PAINT          1         0         0         30       1.0000          3.0
( 3) STAMP          1         0         0         30       1.0000          2.0
( 4) CRIMP          1         0         0         30       1.0000          3.0

        2 TOTAL FAILURES REQUIRED      2 INTERVALS REPAIR TIME
       AVERAGE REPAIR TIME=    1.00

       SYSTEM WAS DEFECT-FREE FOR    28 INTERVALS OUT OF   30
       DEMAND FOR MAINTENANCE ACTION= .0667
```

Figure 20 — Example 3.2 — Page 3

Ending this section of the output are a couple of lines relating repair and maintenance information. A system summary of failures and repair time gives an estimate of average repair time. Lastly, an account is given of how often the system began an interval with no machines down. The number of remaining intervals over the total simulation time indicates the percentage of time that the presence of a repairman was required, here 6.67%. Notice that GENMOD deliberately does not label these two statistics as "system MTTR" and "system non-availability" respectively. This is done so as not to enter into the politics of what really constitutes a system.

Because the variable IREW2 was set at 2 on the EXECUTE card for this run, the buffer status tape was rewound and copied onto the output at this point. Furthermore, the option (2) used here results in the most basic form of this table, namely a single line that relates the contents of the buffers selected (by placing a "1" for print code on the buffer card) as of the end of the interval shown. This particular printout can be followed rather easily. It readily shows the passage of parts into the output buffer, beginning after the fourth interval, the gradual buildup in Buffers 1, 2, and 3 due to the differences in rates, and the decreases in Buffer 1 that accompany the two failures of Machine 1.

Note that after the 29th interval, Buffer 2 contains 289 parts, just one shy of its maximum capacity. In the next interval, if all went well, Machine 3 would remove 66 parts since it was not scheduled to go down nor was its output buffer in danger of being filled. This is indeed what happened, as seen on the machine handling table. This removal of 66 parts left room for only 67 new parts in Buffer 2. That explains the 67 that Machine 2 was shown as handling in the 29th interval. In the ensuing interval, with Buffer 2 already full, all that could be put into it would be the same as what is removed, namely another 66 by Machine 3. Thus, Machine 2 only handled 66 pencils in the 30th interval. It would in fact have continued handling 66 parts per intervals until at least interval 41 (when Machine 3 was scheduled to go down) had it not gone down itself in interval 36.

Following the last entry in the Buffer status table is a quick check of the maximum material ever contained in those buffers. This is useful here since these are usually the main buffers of interest, with maximum capacity being a major characteristic sought, and because the buffer status is not usually printed every single interval, so that the maximum could very likely have been achieved during an interval that is not printed out. The maximum in the output buffer is, of course, the total finished product released by the model during the run. It is advisable, though hardly necessary, that the output point be included as one of the buffers to be printed on the status table.

The other forms of the periodic buffer status table will be discussed through later examples.

We have finally reached one of the most important areas of the printout, that showing the expected output from the model. The first average, per basic time interval, is quite simply the total output (e.g. 1620) divided by the total simulation time (e.g. 30). It is clear, then, that GENMOD is only interested in the finished product and how long it took to get that way. The reasoning behind this is that the importance placed upon semi-processed parts is only subjective and because, with variable buffer seeding available, the only common

```
                 ACME PENCIL FINISHING LINE
                 PERIODIC BUFFER STATUS
                 -----------------------

INTERVAL        1        2        3        4
--------------------------------------------
    1          83.       0.       0.       0.
--------------------------------------------
    2          91.      74.       0.       0.
--------------------------------------------
    3          99.      83.      66.       0.
--------------------------------------------
    4         107.      91.      72.      60.
--------------------------------------------
    5         116.      99.      78.     120.
--------------------------------------------
    6         124.     107.      84.     180.
--------------------------------------------
    7         132.     116.      90.     240.
--------------------------------------------
    8         140.     124.      96.     300.
--------------------------------------------
    9         149.     132.     102.     360.
--------------------------------------------
   10         157.     140.     108.     420.
--------------------------------------------
   11         126.     149.     114.     480.
--------------------------------------------
   12         125.     157.     120.     540.
--------------------------------------------
   13         134.     165.     126.     600.
--------------------------------------------
   14         142.     173.     132.     660.
--------------------------------------------
   15         150.     182.     138.     720.
--------------------------------------------
   16         158.     190.     144.     780.
--------------------------------------------
   17         167.     198.     150.     840.
--------------------------------------------
   18         175.     206.     156.     900.
--------------------------------------------
   19         183.     215.     162.     960.
--------------------------------------------
   20         182.     223.     168.    1020.
--------------------------------------------
   21         125.     231.     174.    1080.
--------------------------------------------
   22         134.     239.     180.    1140.
--------------------------------------------
   23         142.     248.     186.    1200.
--------------------------------------------

INTERVAL        1        2        3        4
```

Figure 21 — Example 3.2 — Page 4

```
          ACME PENCIL FINISHING LINE

          PERIODIC BUFFER STATUS
          ----------------------

INTERVAL          1       2       3       4
----------------------------------------------
   24           150.    256.    192.   1260.
----------------------------------------------
   25           158.    264.    198.   1320.
----------------------------------------------
   26           167.    272.    204.   1380.
----------------------------------------------
   27           175.    281.    210.   1440.
----------------------------------------------
   28           183.    289.    216.   1500.
----------------------------------------------
   29           198.    290.    222.   1560.
----------------------------------------------
   30           215.    290.    228.   1620.

MAXIMUM
MATERIAL
HANDLED-        215.    290.    228.   1620.

PROJECTED AVERAGE OUTPUTS
-------------------------
          PER BASIC INTERVAL-            54.00 (    54.00 PER MINUTE)
          PER 420.-INTERVAL SHIFT-    22680.
          PER 3/8/5 WEEK         -   340200.
          PER 63-SHIFT MONTH   -    1428840.
```

Figure 22 — Example 3.2 — Page 4 (Cont'd)

time factor is total simulation time. Any inferences the user wishes to place upon semi-processed parts are always available to him through buffer analysis. Likewise, the user is free to adjust the projected output to meet his particular definition. In particular, recalling that Machine 4, the last step in our model, was not really producing for the first three intervals, one approach might be to divide the output by 27 rather than 30. This would yield a figure of 60, rather than 54, and reflects what the last machine did indeed do. But, does it represent what the model did?

Since one minute is by no means the only choice available for BTI, and since Parts-per-minute, though, is a common unit for expressing output, the program converts the BTI output to PPM. In this instance, of course, these two values are the same.

The three remaining lines are simply extrapolations of this basic rate. An estimate is made of shift output, based upon the user's specification of same. This is in fact the only place that the variable SHIFTL is used. For this example, we input a shift length of 420 minutes, yielding an estimate of 23680 pencils. For convenience, this shift output is extended to a full five-day week of three such shifts daily. Here that value is 340,200. Finally, the consequence of this model over a typical month of 21 days with three shifts each are shown. In many cases, these lines will be of no interest to the user. They are there solely for the convenience of those who can use them.

Even though only 15 buffers can be printed out on the status table, GENMOD nonetheless keeps track of all buffers. In the next section of the standard printout, the maximum, average, and final content for all buffers are listed. This section will be printed regardless of the option used for the periodic buffer status table. For the output point, the maximum and final are always the same and the average value is of no consequence.

The last parameter on the EXECUTE card, IREW3, controls printing of the system logbook, that details each failure and repair on the line. The only options available here are to print the logbook or not.

Whenever a machine goes down, the time is shown in the following format: XX.YY where XX is the interval in which the failure occurred, and YY is how far into that interval the failure actually took place. Thus, for Machine 1's first failure, 11.53 means a little beyond halfway through the 11th interval. Along with the failure time is shown the expected uptime as determined by sampling a repair time from the repair distribution, rounding it to a minimum of 1, and adding it to the failure interval number. Thus, the first repair for Machine 1 was due to take place in the very next interval. There is no way we will ever know whether this 1-interval repair time represented a sampled value that was greater than 1 and then truncated or was less than 1 and rounded up. That information is lost forever.

```
MAXIMUM BUFFER CONTENTS
-----------------------
 (    1=     215.)  (    2=      290.)  (    3=      228.)  (    4=     1620.)

AVERAGE BUFFER CONTENTS
-----------------------
 (    1=     146.)  (    2=      183.)  (    3=      137.)  (    4=      756.)

  FINAL BUFFER CONTENTS
  ---------------------
 (    1=     215.)  (    2=      290.)  (    3=      228.)  (    4=     1620.)
```

Figure 23 — Example 3.2 — Page 5

```
          ACME PENCIL FINISHING LINE

          SYSTEM LOG
          ----------

INT=    11.53  MACHINE  1 DOWN. DUE BACK UP AT   12. TRES=       405.
INT=    12.89  MACHINE  1 BACK UP. NEXT FAILURE AT    20
INT=    20.89  MACHINE  1 DOWN. DUE BACK UP AT   21. TRES=       560.
INT=    21.21  MACHINE  1 BACK UP. NEXT FAILURE AT    49
```

Figure 24 — Example 3.2 — Logbook

For the record, an expected repair time of one interval could actually represent a period ranging from 0.0 to 2.0 intervals. Similarly, any expected repair time will result in a repair taking place anywhere within that time ±1 interval. This can be seen by examining the worst case situations. For example, suppose a failure took place in the 30th interval and is scheduled to last four intervals. The latest that the failure could have taken place was 30.9999...and the earliest the repair could have been made was 34.0000..., for a difference of essentially only three intervals. Similarly, the earliest failure would be 30.0000...with the latest repair being 34.9999..., for a difference of five intervals. This happens for every repair. There is a two-interval range over which it could take place, although the mean of the repairs will be the sampled value. Thus, an expected repair time of one interval means a repair that could take as long as two intervals or could be instantaneous. It should be clear then why a sampled expected repair time must be at least one interval. Otherwise, the repair could take place before the failure.

Regardless of what the numerical differences are between the precise failure and repair times, GENMOD only keeps track of the sampled expected repair times, because these values have a statistical mean of the true sampled repair time. For example, the two repairs of Machine 1 on this run took 1.36 and 0.32 intervals respectively, for an average of 0.84, even though both were slated to take a whole interval each. In the long run, the observed average and the expected average will be the same.

Note carefully that the above discussion only applied to the two-interval range about the sampled repair time within which it could be affected. There is no such limitation about the range of the sampled times themselves. They and the sampled failure times will vary according to their stated distributions. Yet, in the long run, their observed average too will tend toward the input mean. That is why these two repairs of Machine 1 only took an average of one interval each, even though the input MTTR was four. We simply did not run the simulation long enough.

A study will be presented shortly to show the sort of variation to expect in sampled failure and repair times, especially when the exponential distribution is selected rather than the Gaussian.

Going back to the logbook, on the same line with each failure is printed a variable called TRES. This represents the total material resident on the line at the time of the failure. It is useful for gauging the consequencies of possibly invoking the GENMOD option by which all buffers are purged whenever there is a failure.

Proceeding to the logbook line relating to the repair, along with the actual time of the repair is printed the time of the next failure as sampled once again from the failure distribution.

For both failures and repairs, if ever the time should be printed as XX.**, it means the event took place beyond 99% of the way into that interval.

Be aware that in large models involving relatively short failure times, the logbook can get quite lengthy. That is one reason it is printed last, so that the user has a chance to kill the job if the logbook appears too long. Actually, in most cases, the logbook is only useful in debugging models.

This finally concludes the examination of the printout from just one short run of a very simple model. The analysis, though, was deliberately lengthy so as to save time and space on future examples.

3.3 Buffer Change

The expression given to estimate buffer sizes in the preceding case considered the expected backlog due to the exiting machine being down. A similar expression considers the expected backlog due to the entering machine not going down:

Estimated Buffer Size = (Entering Rate – Exiting Rate)
X (MTBF of Entering Machine) X
(Availability of Exiting Machine)

Inflating these estimates by 10% yields a value of 396 for Buffer 2, with the other values being less than what the first expression yielded. Since Buffer 2 did indeed overflow with our initial estimate of 290, we will change our buffer cards to reflect the maximum of these two estimating expressions. At the same time, we shall double the duration of the run from 30 intervals to 60. No other changes have been made to the model or deck.

Upon running this revised model, the first thing we notice on page 1 of the output is the new set of times to first failure. As expected, they are completely different from the initial set. For the sake of comparison, these times are:

Machine 1 = 11, 23	Machine 2 = 36, 62
Machine 3 = 41, 33	Machine 4 = 43, 86

We will come back to these times again later after a couple more runs have been made.

The only other changes seen on this page of the printout are the higher capacity of Buffer 2 (396 vs 290) and the longer simulation time (60 vs 30).

```
0000000000000000000000'
ACME PENCIL FINISHING LINE
004
TRIM        PAINT       STAMP       CRIMP
004
  1  1  1  0  1   82.50   36.00   0.000   4.000   0.000   0.000  1  0
  2  2  1  1  2   74.25   48.00   0.000   6.000   0.000   0.000  1  0
  3  3  1  2  3   66.00   40.00   0.000   4.000   0.000   0.000  1  0
  4  4  1  3  4   60.00   45.00   0.000   5.000   0.000   0.000  1  0
004
  1   0.00000   490.000    1
  2   0.00000   396.000    1
  3   0.00000   330.000    1
  4   0.00000   0.00000    1
   82.5000   420.000    0    0    0    0   1.00000
0000000000000000000000
   60    1    1    2    0
0000000000000000000000
```

Figure 25 — Example 3.3 — Input Deck

When comparing page 2 of the output, Periodic Machine Handling Table, several interesting points come to light.

a. The first four intervals are the same on both runs, i.e. it still takes four intervals to reach steady state. But, in the second example, this steady state condition lasts through Interval 22 because no failures occur or buffers overflow until the 23rd interval. Recall that in the first run, the steady state condition only lasted through the 10th interval.

b. For the first time we notice machines not handling any parts for varying lengths of time. In fact, we see three strings of zero's. The first is attributable to a protracted failure of Machine 1, which went down in Interval 23 and did not come up again until Interval 31. The buildup of excess parts in Buffer 1 was sufficient to allow Machine 2 to continue functioning at full speed for three intervals following the failure. However, in Interval 26 it was forced to slow to 58 parts and then stop itself for five full intervals. Machine 3 was not affected until it had to slow down in Interval 30 and then stop. Notice that Machine 4 never had to slow down or stop at all.

c. The only other interruption to machine flow seen on this page is the short failure and repair of Machine 3 in Intervals 33 and 34 respectively.

```
        ACME PENCIL FINISHING LINE

                          INPUT   OUTPUT             FAILURE        REPAIR      NOMINAL  COMP   AUX   DEFECT
MACHINE  OPERATION(SEQ)  BUFFER  BUFFER   RATE   MEAN  SIGMA   MEAN  SIGMA   AVAIL.   TYPE  FIELD   RATE
-------  --------------  ------  ------   ----   ----  -----   ----  -----   -------  ----  -----  -------
   1     TRIM      ( 1)     0       1     82.5   36.0    0.0    4.0    0.0    .9000     1      0   0.00000
   2     PAINT     ( 1)     1       2     74.3   48.0    0.0    6.0    0.0    .8889     1      0   0.00000
   3     STAMP     ( 1)     2       3     66.0   40.0    0.0    4.0    0.0    .9091     1      0   0.00000
   4     CRIMP     ( 1)     3       4     60.0   45.0    0.0    5.0    0.0    .9000     1      0   0.00000

TIMES OF FIRST FAILURE
----------------------
 (  1=    23)   (  2=    62)   (  3=    33)   (  4=    86)

CONTROL POINT DATA
------------------

CONTROL   INITIAL    MAXIMUM    STATUS                      MAXIMUM   MAXIMUM
 POINT    CONTENTS   CAPACITY   PRINT    ENTRIES   EXITS     INPUT     OUTPUT
-------   --------   --------   ------   -------   -----    -------   -------
   1          0.        490.     YES        1        1        82.5      74.3
   2          0.        396.     YES        1        1        74.3      66.0
   3          0.        330.     YES        1        1        66.0      60.0
   4          0.     ********    YES        1        0        60.0       0.0   (OUTPUT POINT)

        RAW MATERIAL AVAILABLE=       82.50 PER INTERVAL
        SHIFT LENGTH=                420.00 INTERVALS
        LENGTH OF SIMULATION=         60    INTERVALS
        BASIC TIME INTERVAL=           1.00 MINUTES
-----------------------------------------------------------------------------------------------------------
```

Figure 26 — Example 3.3 — Page 1

```
                ACME PENCIL FINISHING LINE

                PERIODIC MACHINE HANDLING
                -------------------------
                   1         2         3         4
       1          83.        0.        0.        0.
       2          83.       74.        0.        0.
       3          83.       74.       66.        0.
       4          83.       74.       66.       60.
       5          83.       74.       66.       60.
       6          83.       74.       66.       60.
       7          83.       74.       66.       60.
       8          83.       74.       66.       60.
       9          83.       74.       66.       60.
      10          83.       74.       66.       60.
      11          83.       74.       66.       60.
      12          83.       74.       66.       60.
      13          83.       74.       66.       60.
      14          83.       74.       66.       60.
      15          83.       74.       66.       60.
      16          83.       74.       66.       60.
      17          83.       74.       66.       60.
      18          83.       74.       66.       60.
      19          83.       74.       66.       60.
      20          83.       74.       66.       60.
      21          83.       74.       66.       60.
      22          83.       74.       66.       60.
      23          25.       74.       66.       60.
      24           0.       74.       66.       60.
      25           0.       74.       66.       60.
      26           0.       58.       66.       60.
      27           0.        0.       66.       60.
      28           0.        0.       66.       60.
      29           0.        0.       66.       60.
      30           0.        0.       58.       60.
      31          58.        0.        0.       60.
      32          83.       58.        0.       60.
      33          83.       74.       33.       60.
      34          83.       74.       62.       60.
      35          83.       74.       66.       60.
      36          83.       74.       66.       60.
      37          83.       74.       66.       60.
      38          83.       74.       66.       60.
      39          83.       74.       66.       60.
      40          83.       74.       66.       60.
      41          83.       74.       66.       60.
      42          83.       74.       66.       60.
      43          83.       74.       66.       60.
      44          83.       74.       66.       60.
      45          83.       74.       66.       60.
                   1         2         3         4
```

Figure 27 — Example 3.3 — Page 2

```
          ACME PENCIL FINISHING LINE

          PERIODIC MACHINE HANDLING
          -------------------------
                 1      2      3      4
 46            83.    74.    66.    60.
 47            83.    74.    66.    60.
 48            83.    74.    66.    60.
 49            83.    74.    66.    60.
 50            83.    74.    66.    60.
 51            83.    74.    66.    60.
 52            83.    74.    66.    60.
 53            83.    74.    66.    60.
 54            83.    74.    66.    60.
 55            83.    74.    66.    60.
 56            83.    74.    66.    60.
 57            83.    74.    66.    60.
 58            83.    74.    66.    60.
 59             6.    74.    66.    60.
 60             0.    74.    66.    60.
TOTAL PARTS HANDLED BY EACH MACHINE
-----------------------------------
 (  1=     4131.)  (  2=     3977.)  (  3=     3651.)  (  4=     3420.

TOTAL RAW MATERIAL CONSUMED=     4131.
TOTAL MATERIAL IN-PROCESS=        711.
```

Figure 28 – Example 3.3 – Page 2 (Cont'd)

The latter portion of the machine handling table reveals one more failure of Machine 1 in the 59th interval.

Since this run was for twice as long as the initial run, we note that the total parts handled by each machine are indeed about double the original figures. The total raw material consumed, though, is somewhat less than double, due to the longer downtimes of Machine 1.

An interesting comparison is of total material still in-process. Even though the second run was for twice as long, used a higher total buffer capacity, and there was never a slowdown due to full buffers, there is still less material sitting around in the buffers than the first go around. This, of course, is the result of several machines having to live off their buffers for several intervals.

```
AVAILABILITY DATA          (BY MACHINE )
-----------------

                                TOTAL     TOTAL      TOTAL     AVERAGE        S T O P    T I M E S
MACHINE  OPERATION(SEQ)        FAILURES  DOWN TIME  UP TIME  AVAILABILITY   DOWNSTREAM  UPSTREAM   TOTAL
-------  -------------         --------  ---------  -------  ------------   ----------  --------   -----
   1     TRIM        ( 1)         2          9        51        .8500           0           0        0
   2     PAINT       ( 1)         0          0        60       1.0000           0           6        6
   3     STAMP       ( 1)         1          1        59        .9833           0           4        4
   4     CRIMP       ( 1)         0          0        60       1.0000           0           3        3

TOTAL UNITS REJECTED=       0.

AVAILABILITY DATA          (BY CATEGORY)
-----------------

                   NO. OF     TOTAL     TOTAL      TOTAL     AVERAGE      AVERAGE
      CATEGORY    MACHINES  FAILURES  DOWN TIME  UP TIME  AVAILABILITY  STOP TIME
      --------    --------  --------  ---------  -------  ------------  ---------
( 1) TRIM             1         2         9        51        .8500         0.0
( 2) PAINT            1         0         0        60       1.0000         6.0
( 3) STAMP            1         1         1        59        .9833         4.0
( 4) CRIMP            1         0         0        60       1.0000         3.0

          3 TOTAL FAILURES REQUIRED       10 INTERVALS REPAIR TIME
        AVERAGE REPAIR TIME=     3.33

        SYSTEM WAS DEFECT-FREE FOR     50 INTERVALS OUT OF     60
        DEMAND FOR MAINTENANCE ACTION= .1667
```

Figure 29 — Example 3.3 — Page 3

ACME PENCIL FINISHING LINE

PERIODIC BUFFER STATUS

INTERVAL	1	2	3	4
1	83.	0.	0.	0.
2	91.	74.	0.	0.
3	99.	83.	66.	0.
4	107.	91.	72.	60.
5	116.	99.	78.	120.
6	124.	107.	84.	180.
7	132.	116.	90.	240.
8	140.	124.	96.	300.
9	149.	132.	102.	360.
10	157.	140.	108.	420.
11	165.	149.	114.	480.
12	173.	157.	120.	540.
13	182.	165.	126.	600.
14	190.	173.	132.	660.
15	198.	182.	138.	720.
16	206.	190.	144.	780.
17	215.	198.	150.	840.
18	223.	206.	156.	900.
19	231.	215.	162.	960.
20	239.	223.	168.	1020.
21	248.	231.	174.	1080.
22	256.	239.	180.	1140.
INTERVAL	1	2	3	4

Figure 30 — Example 3.3 — Page 4

ACME PENCIL FINISHING LINE

PERIODIC BUFFER STATUS

INTERVAL	1	2	3	4
23	206.	248.	186.	1200.
24	132.	256.	192.	1260.
25	58.	264.	198.	1320.
26	0.	256.	204.	1380.
27	0.	190.	210.	1440.
28	0.	124.	216.	1500.
29	0.	58.	222.	1560.
30	0.	0.	220.	1620.
31	58.	0.	160.	1680.
32	83.	58.	100.	1740.
33	91.	99.	72.	1800.
34	99.	111.	75.	1860.
35	107.	120.	81.	1920.
36	116.	128.	87.	1980.
37	124.	136.	93.	2040.
38	132.	144.	99.	2100.
39	140.	153.	105.	2160.
40	149.	161.	111.	2220.
41	157.	169.	117.	2280.
42	165.	177.	123.	2340.
43	173.	186.	129.	2400.
44	182.	194.	135.	2460.
INTERVAL	1	2	3	4

Figure 31 — Example 3.3 — Page 4 (Cont'd)

```
          ACME PENCIL FINISHING LINE

          PERIODIC BUFFER STATUS
          ----------------------

INTERVAL          1        2      3       4
----------------------------------------------
   45            190.     202.   141.   2520.
----------------------------------------------
   46            198.     210.   147.   2580.
----------------------------------------------
   47            206.     219.   153.   2640.
----------------------------------------------
   48            215.     227.   159.   2700.
----------------------------------------------
   49            223.     235.   165.   2760.
----------------------------------------------
   50            231.     243.   171.   2820.
----------------------------------------------
   51            239.     252.   177.   2880.
----------------------------------------------
   52            248.     260.   183.   2940.
----------------------------------------------
   53            256.     268.   189.   3000.
----------------------------------------------
   54            264.     276.   195.   3060.
----------------------------------------------
   55            272.     285.   201.   3120.
----------------------------------------------
   56            281.     293.   207.   3180.
----------------------------------------------
   57            289.     301.   213.   3240.
----------------------------------------------
   58            297.     309.   219.   3300.
----------------------------------------------
   59            229.     318.   225.   3360.
----------------------------------------------
   60            154.     326.   231.   3420.

MAXIMUM
MATERIAL
HANDLED-         297.     326.   231.   3420.

PROJECTED AVERAGE OUTPUTS
-------------------------
         PER BASIC INTERVAL-             57.00(  57.00 PER MINUTE)
         PER 420.-INTERVAL SHIFT-    23940.
         PER 3/8/5 WEEK          -  359100.
         PER 63-SHIFT MONTH    -   1508220.
```

Figure 32 — Example 3.3 — Page 4 (Cont'd)

```
MAXIMUM BUFFER CONTENTS
 (  1=       297.)  (  2=       326.)  (  3=       231.)  (  4=      3420.)
AVERAGE BUFFER CONTENTS
 (  1=       161.)  (  2=       179.)  (  3=       143.)  (  4=      1653.)
 FINAL BUFFER CONTENTS
 (  1=       154.)  (  2=       326.)  (  3=       231.)  (  4=      3420.)
```

Figure 33 — Example 3.3 — Page 5

The figures in the availability summary tables accurately reflect what happened on this run. There was one more total failure, resulting in a total of eight more intervals of repair time. This raised the average repair time to 3.33 intervals and increased the demand for maintenance action to 16.67%. No downstream stoptime was experienced anywhere, but there was more than twice as much upstream stoptime. Once again, of course, there were no units rejected.

The Periodic Buffer Status Table is presented with virtually no comment. When viewed in conjunction with the machine handling table it is fairly easy to follow the flow of pencils through the buffers. Note, however, the periods when Buffers 1 and 2 actually ran dry.

As for maximum material ever contained in each of the buffers, they were all well below the capacities placed upon them for this run.

Despite the increased total downtime and the lower total material available in the buffers, the projected average output for this run actually increased from 54 to 57 PPM. This is due, of course, purely to the fact that Machine 4 has not yet been observed to have failed and was always operating at 60 pencils per interval for all but the first three intervals of both of these runs.

Nothing much different is apparent on the fifth page of the output. The maximum buffer sizes were all higher, but none reached full capacity. The averages were about the same, except for Buffer 1, which was slightly higher. The final values were slightly higher than on the first run, except again for Buffer 1 which ran dry for such a long period of time before refilling.

The logbook contains nothing much interesting except for the hint that at one time there was more material (825) resident in the system then was apparent from any previously printed statistic. In general, from now on, the logbook will not be discussed for any example and in many cases will not even be run.

```
          ACME PENCIL FINISHING LINE

          SYSTEM LOG
          ----------

INT=    23.30  MACHINE  1 DOWN. DUE BACK UP AT   31. TRES=       675.
INT=    31.70  MACHINE  1 BACK UP. NEXT FAILURE AT    59
INT=    33.57  MACHINE  3 DOWN. DUE BACK UP AT   34. TRES=       240.
INT=    34.94  MACHINE  3 BACK UP. NEXT FAILURE AT   127
INT=    59.07  MACHINE  1 DOWN. DUE BACK UP AT   63. TRES=       825.
```

Figure 34 – Example 3.3 – Logbook

3.4 <u>Reject Rates and Delays</u>

We are now ready to incorporate a couple of additional features into our basic model. Let us suppose that the trimming operation has built into its exit conveyor a pair of GO/NOGO gauges by which both short and long pencils get rejected. The short ones must be dumped, but the long ones can ultimately be recycled. We won't yet worry about recycling, just rejecting. We will say that experience has indicated a total reject rate from this operation of 1%. Thus, we will enter a defect rate for Machine 1 of 0.010.

Next, recall the assumption made about the paint being applied and dried all in one interval. We will now adjust our model to reflect a true drying time of, say, three minutes. In other words, we will hypothesize a conveyor behind Machine 2 that takes three minutes to traverse, thus <u>delaying</u> the arrival of painted pencils into Buffer 2 for a 3-interval period. This is an ideal situation to use GENMOD's delay option for at least two reasons.

a. The spraying itself happens fast enough that we can still think of it as a simple operation rather than some sort of batch process that takes three intervals.

b. The painted pencils will be spaced along this conveyor in quantities that reflect the rate at which they were sprayed sometime before. If the painter goes down, the conveyor stops moving. We here have the opportunity to incorporate a perfect conveyor without having to add another machine to the model.

To add this delay time, all we have to do is change the AUX field for Machine 2 from 0 to 3.

Since this is to be only a demonstration run again, we will cut back the simulation time to 30 intervals to save space.

The changes that we have made to the model are readily apparent on the first page of the printout, where the inputs are summarized. Machine 1 does indeed now have a 1% defect rate and the 3-interval delay time for Machine 2 shows up under the AUX FIELD heading.

We note another set of first failure times, ranging from 2 to 29. Comparing the sets for each machine, after three runs now, we see the following:

1 = (11, 23, 2)	2 = (36, 62, 10)
3 = (41, 33, 29)	4 = (43, 86, 16)

```
0000000000000000000000
ACME PENCIL FINISHING LINE
004
TRIM      PAINT      STAMP      CRIMP
004
  1  1  1  0  1   82.50   36.00   0.000   4.000   0.000   0.010  1  0
  2  2  1  1  2   74.25   48.00   0.000   6.000   0.000   0.000  1  3
  3  3  1  2  3   66.00   40.00   0.000   4.000   0.000   0.000  1  0
  4  4  1  3  4   60.00   45.00   0.000   5.000   0.000   0.000  1  0
004
  1   0.00000   490.000   1
  2   0.00000   396.000   1
  3   0.00000   330.000   1
  4   0.00000   0.00000   1
   82.5000   420.000   0    0    0    0   1.00000
0000000000000000000000
   30   1   1   2   0
0000000000000000000000
```

Figure 35 — Example 3.4 — Input Deck

```
         ACME PENCIL FINISHING LINE

                               INPUT   OUTPUT                FAILURE          REPAIR       NOMINAL   COMP    AUX    DEFECT
MACHINE    OPERATION(SEQ)     BUFFER   BUFFER     RATE     MEAN   SIGMA    MEAN   SIGMA    AVAIL.    TYPE   FIELD    RATE
-------    --------------     ------   ------     ----     ----   -----    ----   -----   -------    ----   -----   ------
   1       TRIM       ( 1)       0        1       82.5     36.0    0.0      4.0    0.0     .9000      1       0      .01000
   2       PAINT      ( 1)       1        2       74.3     48.0    0.0      6.0    0.0     .8889      1       3     0.00000
   3       STAMP      ( 1)       2        3       66.0     40.0    0.0      4.0    0.0     .9091      1       0     0.00000
   4       CRIMP      ( 1)       3        4       60.0     45.0    0.0      5.0    0.0     .9000      1       0     0.00000
TIMES OF FIRST FAILURE
----------------------
  (  1=      2)  (  2=      10)  (  3=      29)  (  4=      16)

CONTROL POINT DATA
------------------

CONTROL   INITIAL    MAXIMUM    STATUS                    MAXIMUM   MAXIMUM
 POINT    CONTENTS   CAPACITY   PRINT   ENTRIES  EXITS    INPUT     OUTPUT
-------   --------   --------   ------  -------  -----   -------   -------
   1         0.         490.     YES       1       1      82.5      74.3
   2         0.         396.     YES       1       1      74.3      66.0
   3         0.         330.     YES       1       1      66.0      60.0
   4         0.     ********     YES       1       0      60.0       0.0   (OUTPUT POINT)

       RAW MATERIAL AVAILABLE=        82.50 PER INTERVAL
       SHIFT LENGTH=                 420.00 INTERVALS
       LENGTH OF SIMULATION=          30    INTERVALS
       BASIC TIME INTERVAL=            1.00 MINUTES

------------------------------------------------------------------------------------------------------------------------
```

Figure 36 — Example 3.4 — Page 1

The periodic machine handling table for this run contains some very interesting items, as all machines actually failed, but only once and with no overlaps.

a. The failure of Machine 1 is clearly seen in Interval 2, with only 29 parts being processed in that interval and none for the next three. This stoppage of the first machine is felt on the very next interval by Machine 2, which only had available to it 36 parts and then had to stop itself for three intervals.

b. Hithertofore, Machine 3 always got to handle pencils in the third interval. Notice that here, though, it did not process anything until the sixth interval, three later than normal. This is a direct consequence of tacking on a 3-interval delay behind Machine 2.

c. Because of the 3-interval span of no production out of Machine 1, this run nicely shows the passage through the line of the first two minutes' worth of production. This starts out as about 112 parts, of which approximately 1% is rejected. The remainder, about 111 parts, then appears as 74 + 36, 66 + 45, and 60 + 51 for Machines 2, 3 and 4, respectively. (Remember that the values on the printout are always rounded, accounting for minor variations in totals. Also, identical values do not always appear the same. Notably, the ideal production for Machine 1 is 82.5 PPM, yet this appears as both 83. and 82. on the table.)

d. A complex situation is seen beginning around Interval 10. The 43 parts handled by Machine 2 in the seventh interval is clearly the same 43 parts handled by Machine 3 in Interval 11. But, where did the 42 and 61 parts come from for the next two intervals? We note that Machine 2 handled 61 and 42 during Intervals 10 and 11. On the surface it would appear that not only did the delay function misfire, but that the order of parts was also reversed. What exactly happened here? To find out, we must thoroughly understand how the delay function works and we must also reconstruct part of the buffer status table, including the three cells on the conveyor between Machines 2 and 3. This table appears below, beginning at Interval 6.

Interval	Buffer 1	Cell 1	Cell 2	Cell 3	Buffer 2	Buffer 3	Buffer 4
6	43	0	0	0	45	66	0
7	82	43	0	0	0	51	60
8	89	74	43	0	0	0	111
9	97	74	74	43	0	0	111
10	117	74	74	61	43	0	111
11	157	74	74	61	42	43	111
12	165	74	74	74	61	42	154
13	172	74	74	74	74	61	196

TABLE 6

```
        ACME PENCIL FINISHING LINE

        PERIODIC MACHINE HANDLING
        -------------------------
                  1      2      3      4
     1           83.     0.     0.     0.
     2           29.    74.     0.     0.
     3            0.    36.     0.     0.
     4            0.     0.     0.     0.
     5            0.     0.     0.     0.
     6           44.     0.    66.     0.
     7           83.    43.    45.    60.
     8           82.    74.     0.    51.
     9           83.    74.     0.     0.
    10           83.    61.     0.     0.
    11           82.    42.    43.     0.
    12           83.    74.    42.    43.
    13           83.    74.    61.    42.
    14           83.    74.    66.    60.
    15           83.    74.    66.    60.
    16           83.    74.    66.    37.
    17           82.    74.    66.     6.
    18           83.    74.    66.    60.
    19           83.    74.    66.    60.
    20           83.    74.    66.    60.
    21           83.    74.    66.    60.
    22           83.    74.    66.    60.
    23           83.    74.    66.    60.
    24           83.    74.    66.    60.
    25           83.    74.    66.    60.
    26           83.    74.    66.    60.
    27           83.    74.    66.    60.
    28           83.    74.    66.    60.
    29           82.    74.    55.    60.
    30           83.    74.     0.    60.

TOTAL PARTS HANDLED BY EACH MACHINE
-----------------------------------
 (  1=     2135.)  (  2=     1816.)  (  3=     1302.)  (  4=     1138.)

TOTAL RAW MATERIAL CONSUMED=     2135.
TOTAL MATERIAL IN-PROCESS=        977.
```

Figure 37 — Example 3.4 — Page 2

```
AVAILABILITY DATA          (BY MACHINE )
-----------------

                          TOTAL     TOTAL      TOTAL      AVERAGE       S T O P   T I M E S
MACHINE  OPERATION(SEQ)  FAILURES  DOWN TIME  UP TIME  AVAILABILITY  DOWNSTREAM  UPSTREAM  TOTAL
-------  --------------  --------  ---------  -------  ------------  ----------  --------  -----
   1     TRIM      ( 1)      1         4         26       .8667           0          0       0
   2     PAINT     ( 1)      1         1         29       .9667           0          4       4
   3     STAMP     ( 1)      1         1         29       .9667           0          8       8
   4     CRIMP     ( 1)      1         1         29       .9667           0          9       9

TOTAL UNITS REJECTED=    20.

AVAILABILITY DATA              (BY CATEGORY)
-----------------
                        NO. OF    TOTAL     TOTAL      TOTAL      AVERAGE      AVERAGE
         CATEGORY      MACHINES  FAILURES  DOWN TIME  UP TIME  AVAILABILITY  STOP TIME
         --------      --------  --------  ---------  -------  ------------  ---------
     ( 1) TRIM             1         1         4         26        .8667        0.0
     ( 2) PAINT            1         1         1         29        .9667        4.0
     ( 3) STAMP            1         1         1         29        .9667        8.0
     ( 4) CRIMP            1         1         1         29        .9667        9.0

               4 TOTAL FAILURES REQUIRED        7 INTERVALS REPAIR TIME
              AVERAGE REPAIR TIME=    1.75

              SYSTEM WAS DEFECT-FREE FOR    23 INTERVALS OUT OF    30
              DEMAND FOR MAINTENANCE ACTION= .2333
```

Figure 38 — Example 3.4 — Page 3

ACME PENCIL FINISHING LINE

PERIODIC BUFFER STATUS

INTERVAL	1	2	3	4
1	82.	0.	0.	0.
2	36.	0.	0.	0.
3	0.	0.	0.	0.
4	0.	0.	0.	0.
5	0.	74.	0.	0.
6	43.	45.	66.	0.
7	82.	0.	51.	60.
8	89.	0.	0.	111.
9	97.	0.	0.	111.
10	117.	43.	0.	111.
11	157.	42.	43.	111.
12	165.	61.	42.	154.
13	172.	74.	61.	196.
14	179.	83.	67.	256.
15	187.	91.	73.	316.
16	194.	99.	103.	352.
17	202.	107.	163.	358.
18	209.	116.	169.	418.
19	217.	124.	175.	478.
20	224.	132.	181.	538.
21	232.	140.	187.	598.
22	239.	149.	193.	658.
INTERVAL	1	2	3	4

Figure 39 — Example 3.4 — Page 4

```
         ACME PENCIL FINISHING LINE

         PERIODIC BUFFER STATUS
         ----------------------

INTERVAL            1        2        3        4
----------------------------------------------------
   23              247.     157.     199.     718.
----------------------------------------------------
   24              254.     165.     205.     778.
----------------------------------------------------
   25              262.     173.     211.     838.
----------------------------------------------------
   26              269.     182.     217.     898.
----------------------------------------------------
   27              276.     190.     223.     958.
----------------------------------------------------
   28              284.     198.     229.    1018.
----------------------------------------------------
   29              291.     217.     224.    1078.
----------------------------------------------------
   30              299.     291.     164.    1138.

MAXIMUM
MATERIAL
HANDLED-           299.     291.     229.    1138.


PROJECTED AVERAGE OUTPUTS
-------------------------
         PER BASIC INTERVAL-                  37.95 (     37.95 PER MINUTE)
         PER 420.-INTERVAL SHIFT-     15939.
         PER 3/8/5 WEEK          -   239078.
         PER 63-SHIFT MONTH   -     1004129.
```

Figure 40 — Example 3.4 — Page 4 (Cont'd)

The 43 parts can be seen making their way through the line, via the conveyor, from Buffer 2 to the final output. Note that in Intervals 8 and 9, as the 43 was shifted one cell to the right on the conveyor, 74 new parts were placed on, since both Machines 1 and 2 were still working perfectly. In the tenth interval, however, Machine 2 failed, but not until it had time to process 61 parts. While these parts were being made, the conveyor was still moving, but then it stopped. Since 61 is greater than 43, by the time the failure occurred, all 43 parts from Cell 3 had been deposited into Buffer 2 and were replaced by 61 parts from Cell 2. This latter cell appears unchanged, but in reality, 61 parts were removed and 61 more put in. Likewise with Cell 1. In the eleventh interval, Machine 2 was restored to service in time to process 42 parts. The conveyor did not start moving again until Machine 2 did indeed come back up. Thus, what really happened was that 42 new parts were placed in Cell 1, which moved 42 into Cell 2, which moved 42 into Cell 3, which finally dumped 42 into Buffer 2, now emptied of parts by Machine 3. In the twelfth interval, everything is back to normal, with 74 parts now being placed in Cell 1. This eventually forces all 61 parts from Cell 3 into Buffer 2 and results in all three cells on the conveyor being full.

The last line on this page of the output shows 977 parts still in-process at the end of the run. Yet, the final buffer contents (Figure 41) shows Buffers 1, 2, and 3 containing a total of only 754. The difference between the two is accounted for by the last three intervals of production by Machine 2. This material is located on the conveyor on its way to Buffer 2 and is indeed still in-process. Whenever a delay is used in a model, the parts being delayed will not show up in any buffer, but are included in counts of material in-process or resident, e.g. TRES on the failure lines in the logbook.

```
MAXIMUM BUFFER CONTENTS
-----------------------
(  1=        299.)   ( 2=        291.)  (  3=        229.)  (  4=       1138.)

AVERAGE BUFFER CONTENTS
-----------------------
(  1=        170.)   ( 2=         98.)  (  3=        108.)  (  4=        408.)

FINAL BUFFER CONTENTS
---------------------
(  1=        299.)   ( 2=        291.)  (  3=        164.)  (  4=       1138.)
```

Figure 41 — Example 3.4 — Page 5

The summary page now has an entry under units rejected, i.e. 20, corresponding to the 1% or so that Machine 1 discarded. Do not expect the number of rejects to be precisely the percentage indicated on the data card, as happened here. The number of rejects actually discarded is sampled from a uniform distribution with a mean value of the desired defect rate and a range of 40% of the mean. The average rejected will, of course, be the same as the distribution mean, but the amounts actually discarded each interval will vary. The particular quantities rejected are a percentage of the material actually handled by that machine that interval and are, of course, in the same units. All rejects are lumped together on this line of the printout. So, when different sized parts are being processed and rejected on the same line, there is no way to tell the rejects apart.

Once again, the periodic buffer status table is presented virtually without comment. Its entries can be traced rather easily from the machine handling table, if the 3-interval delay is kept in mind. In particular, the first bit of work done by Machine 2, namely 74 parts in Interval 2, does not show up in Buffer 2 until the end of the fifth interval.

The expected output from this run is considerably lower than the other two, not only because of the delay function that essentially knocks out three intervals of production but also because of the longer duration of the observed failures. The rejected parts also have an effect, but not much. It is useless, though, to worry about the expected output after such a short test run. This line has in no way reached a steady state and this output cannot be used to make any predictions.

From the last page of the output, we see that no buffer ever reached its maximum. This is attributable mostly to the short duration of the run. Under normal conditions, the buffers on this line fill up at six to eight parts per interval, due to the differences in machine speeds, and sooner or later some buffers will get full.

Filling up a buffer, though, is not necessarily a case of the buffer being too small. There may simply be too much material spread out on the line, often as a result of the raw material value being too high. In these runs we fed the head of the line 82.5 pencils per interval merely because that was the maximum speed of the first machine. Many of these parts wound up accumulating in Buffer 1. In other words, we chose to run the first machine at top speed, which in turn caused succeeding machines to operate at top speed, as well as filling up buffers at a predictable rate. If we can meet our desired output rate consistently over long periods (which we have not yet demonstrated), we might very well be able to reduce the raw material flow, thus saving wear and tear on the machines and possibly even reduce buffers. Examples of such line optimization through simulation will be given later.

Once again, the logbook for this run is not even shown.

3.5 Parallel Stations, Contingencies, and Packers

A major expansion shall now be made to our Acme Pencil Line. Suppose we found that no manufacturer makes stamping machines that work as fast as the 66 PPM we want. Instead, we have to install two machines in parallel at 33 PPM each. Also, we found that we can no longer assume an uninterrupted supply of erasers and end caps. So we would like to model the possibility of this supply occasionally stopping. Finally, we don't like the thought of our nice pencils bouncing around the floor after they come out of the crimping machine. To correct this we will add a packaging machine to put the pencils in boxes, eight to a box, at a rate consistent with the rest of the line. The remaining machines and requirements will be the same. The block diagram for our expanded line is shown in Figure 42. Note how I have depicted the eraser supply (Machine 5), as an operation leading nowhere, i.e. to Buffer 999. The supply and crimp blocks will be modeled as a Type 5 group.

The input deck is now getting more complex, as well as more interesting. We've changed the title, added another category and renumbered them, added two more machines and renumbered them, made more use of the TYPE and AUX fields, and added a new buffer. Note in particular the change of type for the dual stampers, from Type 1 to Type 0, the output buffer for Machine 5 (999), the proper use of the Type 5 and -5 codes, and the packaging information in the AUX field of Machine 7.

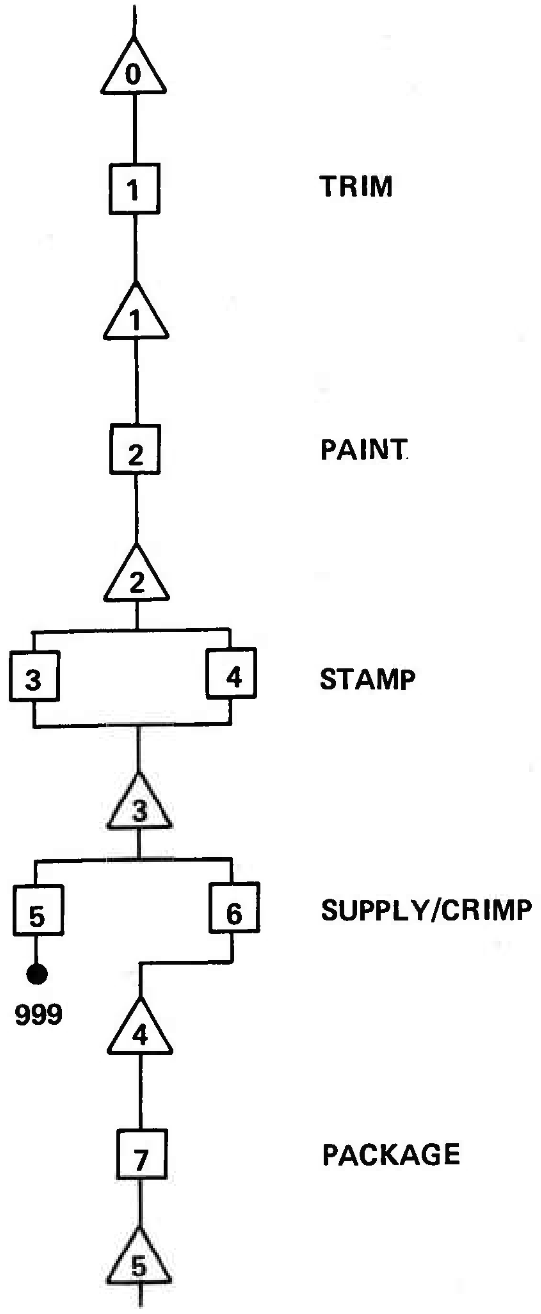


Figure 42

```
0000000000000000000000
ACME LINE WITH SUPPLY
006
TRIM        PAINT       STAMP       SUPPLY      CRIMP       PACKAGE
007
  1   1   1   0   1    82.50    36.00    0.000    4.000    0.000    0.010   1   0
  2   2   1   1   2    74.25    48.00    0.000    6.000    0.000    0.000   1   3
  3   3   1   2   3    33.00    40.00    0.000    4.000    0.000    0.000   0   0
  4   3   2   2   3    33.00    40.00    0.000    4.000    0.000    0.000   0   0
  5   4   1   3999     0.000    50.00    0.000    5.000    0.000    0.000   5   0
  6   5   1   3   4    60.00    45.00    0.000    5.000    0.000    0.000  -5   0
  7   6   1   4   5    64.00    50.00    0.000    6.000    0.000    0.000   7   8
005
  1    0.00000    490.000    1
  2    0.00000    396.000    1
  3    0.00000    330.000    1
  4    0.00000    356.000    1
  5    0.00000    0.00000    1
   82.5000     420.000     0     0     0     0    1.00000
0000000000000000000000
   30     1     1     2     0
0000000000000000000000
```

Figure 43 — Example 3.5 — Input Deck

These many changes are clearly seen on the first page of the printout, with several items not appearing previously.

a. The two stampers are identified by sequence number (1 and 2).

b. The fact that a Type 5 group has been created is indicated. Its last element, the operational element, is indeed Machine 6 and its output does go to Buffer 4.

c. We see for the first time buffers having more than a single entry or exit. The maximum input and output columns are reflective of all machines entering or exiting a buffer. Furthermore, these values are in units that are consistent with the operation going on. In particular, Buffer 4 is shown as having 64 pencils exiting while Buffer 5 only has eight boxes entering.

Note the times to first failure once again, especially Machine 3, slated to fail in the very first interval. This failure time, which came from an exponential distribution with a mean of 40, could very well have actually been less than one, being subsequently rounded up by GENMOD to the minimum manageable time period.

```
        ACME LINE WITH SUPPLY

                                  INPUT   OUTPUT            FAILURE           REPAIR       NOMINAL   COMP    AUX    DEFECT
MACHINE   OPERATION(SEQ)   BUFFER   BUFFER    RATE    MEAN   SIGMA     MEAN   SIGMA    AVAIL.    TYPE   FIELD    RATE
-------   --------------   ------   ------    ----    ----   -----     ----   -----   --------   ----   -----   ------
   1      TRIM      ( 1)      0        1      82.5    36.0     0.0      4.0     0.0     .9000      1      0      .01000
   2      PAINT     ( 1)      1        2      74.3    48.0     0.0      6.0     0.0     .8889      1      3     0.00000
   3      STAMP     ( 1)      2        3      33.0    40.0     0.0      4.0     0.0     .9091      0      0     0.00000
   4      STAMP     ( 2)      2        3      33.0    40.0     0.0      4.0     0.0     .9091      0      0     0.00000
   5      SUPPLY    ( 1)      3      999       0.0    50.0     0.0      5.0     0.0     .9091      5      0     0.00000
   6      CRIMP     ( 1)      3        4      60.0    45.0     0.0      5.0     0.0     .9000     -5      0     0.00000
   7      PACKAGE   ( 1)      4        5      64.0    50.0     0.0      6.0     0.0     .8929      7      8     0.00000

GROUPED COMPONENTS
------------------

          GROUP    LAST ELEMENT    OUTPUT BUFFER
          -----    ------------    -------------
            1           6                4

TIMES OF FIRST FAILURE
----------------------
 (  1=     14)  (  2=      7)  (  3=      1)  (  4=      4)  (  5=    145)  (  6=     79)  (  7=      8)

CONTROL POINT DATA
------------------

CONTROL   INITIAL    MAXIMUM    STATUS                     MAXIMUM   MAXIMUM
 POINT    CONTENTS   CAPACITY   PRINT    ENTRIES   EXITS    INPUT     OUTPUT
-------   --------   --------   ------   -------   -----   -------   -------
   1         0.         490.     YES        1        1       82.5      74.3
   2         0.         396.     YES        1        2       74.3      66.0
   3         0.         330.     YES        2        1       66.0      60.0
   4         0.         356.     YES        1        1       60.0      64.0
   5         0.     ********     YES        1        0        8.0       0.0   (OUTPUT POINT)

        RAW MATERIAL AVAILABLE=        82.50 PER INTERVAL
        SHIFT LENGTH=                 420.00 INTERVALS
        LENGTH OF SIMULATION=          30    INTERVALS
        BASIC TIME INTERVAL=            1.00 MINUTES

------------------------------------------------------------------------------------------------------------------------
```

Figure 44 — Example 3.5 — Page 1

```
          ACME LINE WITH SUPPLY

          PERIODIC MACHINE HANDLING
          -------------------------
                1      2      3      4      5      6      7
    1         83.     0.     0.     0.     0.     0.     0.
    2         83.    74.     0.     0.     0.     0.     0.
    3         83.    74.     0.     0.     0.     0.     0.
    4         83.    74.     0.     0.     0.     0.     0.
    5         83.    74.     0.     0.     0.     0.     0.
    6         83.    74.    33.     0.     0.     0.     0.
    7         83.    19.    33.     0.     0.    33.     0.
    8         83.     0.    33.     0.     0.    33.     9.
    9         83.     0.    33.     0.     0.    33.    40.
   10         83.     0.    33.     0.     0.    33.    50.
   11         64.     0.     3.     0.     0.    33.    33.
   12         60.    59.     0.     0.     0.     3.    33.
   13         75.    74.    33.     0.     0.     0.     3.
   14         66.    74.    33.     0.     0.    33.     0.
   15          0.    74.    33.     0.     0.    33.    33.
   16         83.    74.    33.     0.     0.    33.    33.
   17         83.    74.    33.    13.     0.    33.    33.
   18         83.    74.    33.    33.     0.    46.    33.
   19         83.    74.    33.    33.     0.    60.    46.
   20         83.    74.    33.    33.     0.    60.    60.
   21         83.    74.    33.    33.     0.    60.    60.
   22         83.    68.    33.    33.     0.    60.    60.
   23         25.     0.    33.    33.     0.    60.    60.
   24          0.     0.    33.    33.     0.    60.    60.
   25          0.     0.    33.    33.     0.    60.    60.
   26          0.     0.    32.    33.     0.    60.    60.
   27          0.     0.     0.    25.     0.    60.    60.
   28          0.     0.     0.     0.     0.    60.    60.
   29          0.     0.     0.     0.     0.    18.    60.
   30          0.     0.     0.     0.     0.     0.    18.
```

Figure 45 — Example 3.5 — Page 2 (Partial)

```
AVAILABILITY DATA          (BY MACHINE )
-----------------

                              TOTAL     TOTAL      TOTAL     AVERAGE        S T O P   T I M E S
MACHINE   OPERATION(SEQ)    FAILURES  DOWN TIME   UP TIME  AVAILABILITY  DOWNSTREAM  UPSTREAM  TOTAL
-------   --------------    --------  ---------   -------  ------------  ----------  --------  -----
   1      TRIM       ( 1)       1         1         29        .9667          11          0       11
   2      PAINT      ( 1)       2        13         17        .5667           0          1        1
   3      STAMP      ( 1)       2         5         25        .8333           0          6        6
   4      STAMP      ( 2)       1        13         17        .5667           0          7        7
   5      SUPPLY     ( 1)       0         0         30       1.0000           0          0        0
   6      CRIMP      ( 1)       0         0         30       1.0000           0          8        8
   7      PACKAGE    ( 1)       1         1         29        .9667           0          8        8

TOTAL UNITS REJECTED=    17.

AVAILABILITY DATA          (BY CATEGORY)
-----------------

                     NO. OF     TOTAL      TOTAL      TOTAL     AVERAGE       AVERAGE
      CATEGORY      MACHINES   FAILURES  DOWN TIME   UP TIME  AVAILABILITY   STOP TIME
      --------      --------   --------  ---------   -------  ------------   ---------
( 1)  TRIM              1          1          1         29        .9667        11.0
( 2)  PAINT             1          2         13         17        .5667         1.0
( 3)  STAMP             2          3         18         42        .7000         6.5
( 4)  SUPPLY            1          0          0         30       1.0000         0.0
( 5)  CRIMP             1          0          0         30       1.0000         8.0
( 6)  PACKAGE           1          1          1         29        .9667         8.0

         7 TOTAL FAILURES REQUIRED        33 INTERVALS REPAIR TIME
        AVERAGE REPAIR TIME=     4.71

        SYSTEM WAS DEFECT-FREE FOR      8 INTERVALS OUT OF     30
        DEMAND FOR MAINTENANCE ACTION= .7333
```

Figure 46 — Example 3.5 — Page 3

The periodic machine handling table is indeed getting complicated and contains far too much information to try to cover here. The user is encouraged to track all of these values himself for the experience. To help in this regard, the logbook is presented in full for this case. Armed with the logbook and the periodic buffer status table (Figure 47), all these values should be clear. Of particular interest should be the following items:

a. The fact that Machine 5 (Supply) never "handled" any parts.

b. Machine 7 is rated in terms of pencils, not boxes. All GENMOD machines handle parts in terms of their input. The output from this machine, though, is in terms of boxes.

c. The stamping station only processed 33 parts an interval whenever either machine in the station is down, because 33 is the maximum speed of the remaining machine.

d. The forced slowdown of Machine 1 beginning at Interval 24 due to a full Buffer 1.

e. The wide fluctuation in production by Machine 7, due not so much to its own failure, but to the general lack of parts in its input buffer, reflecting what had happened earlier upstream. As pointed out earlier, such a pattern of operation, exhibited by a machine as a consequence of its interaction with its neighbors, could only be explored through simulation, not just analysis.

The summary page for this run shows for the first time the utility of identifying machines as belonging to categories. Previously there had only been one machine in each category, so that the availability summaries by machine and by category were identical. Here, though, the category table combines the statistics for the two stampers to show how the entire stamping operation performed overall.

Note also the relatively high stoptimes encountered on this run, to the extent that Machine 1 alone had to be slowed down for over one third of the running time. In other words, its availability may have been 97% (29/30), but its efficiency was only 60% (18/30).

One other interesting item here is the demand for maintenance action (73%). As more and more machines are added to a line, the need for repairmen being present generally increases.

Just as with the individual machine handling table, the buffer status table is complicated, but still can be followed. Note the imbalance that exists on this line at present. Buffer 1 is nearly always full because it is being fed at such a high rate, which the successive machines cannot handle. Buffer 4, on the other hand, is nearly always empty, containing in general just what is passed to it in a single interval, due mostly to the stamping operation's inability to keep pace.

```
            ACME LINE WITH SUPPLY

            PERIODIC BUFFER STATUS
            ----------------------

INTERVAL         1        2        3        4        5
-------------------------------------------------------
    1           82.       0.       0.       0.       0.
-------------------------------------------------------
    2           89.       0.       0.       0.       0.
-------------------------------------------------------
    3           97.       0.       0.       0.       0.
-------------------------------------------------------
    4          104.       0.       0.       0.       0.
-------------------------------------------------------
    5          112.      74.       0.       0.       0.
-------------------------------------------------------
    6          119.     116.      33.       0.       0.
-------------------------------------------------------
    7          182.     102.      33.      33.       0.
-------------------------------------------------------
    8          263.      69.      33.      57.       1.
-------------------------------------------------------
    9          345.      36.      33.      50.       6.
-------------------------------------------------------
   10          427.       3.      33.      33.      12.
-------------------------------------------------------
   11          489.       0.       3.      33.      17.
-------------------------------------------------------
   12          489.      59.       0.       3.      21.
-------------------------------------------------------
   13          489.     100.      33.       0.      21.
-------------------------------------------------------
   14          480.     142.      33.      33.      21.
-------------------------------------------------------
   15          407.     183.      33.      33.      25.
-------------------------------------------------------
   16          414.     224.      33.      33.      29.
-------------------------------------------------------
   17          422.     253.      46.      33.      33.
-------------------------------------------------------
   18          429.     261.      66.      46.      37.
-------------------------------------------------------
   19          437.     269.      72.      60.      43.
-------------------------------------------------------
   20          444.     278.      78.      60.      51.
-------------------------------------------------------
   21          452.     286.      84.      60.      58.
-------------------------------------------------------
   22          466.     287.      90.      60.      66.
-------------------------------------------------------

INTERVAL         1        2        3        4        5
```

Figure 47 — Example 3.5 — Page 4

```
            ACME LINE WITH SUPPLY

            PERIODIC BUFFER STATUS
            ----------------------

INTERVAL          1        2        3        4        5
-------------------------------------------------------
   23           490.     221.      96.      60.      73.
-------------------------------------------------------
   24           490.     155.     102.      60.      81.
-------------------------------------------------------
   25           490.      89.     108.      60.      88.
-------------------------------------------------------
   26           490.      24.     113.      60.      96.
-------------------------------------------------------
   27           490.       0.      77.      60.     103.
-------------------------------------------------------
   28           490.       0.      17.      60.     111.
-------------------------------------------------------
   29           490.       0.       0.      17.     118.
-------------------------------------------------------
   30           490.       0.       0.       0.     120.

MAXIMUM
MATERIAL
HANDLED-        490.     287.     113.      60.     120.

PROJECTED AVERAGE OUTPUTS
-------------------------
         PER BASIC INTERVAL-              4.01 (     4.01 PER MINUTE)
         PER 420.-INTERVAL SHIFT-      1684.
         PER 3/8/5 WEEK          -    25262.
         PER 63-SHIFT MONTH      -   106099.
```

Figure 48 — Example 3.5 — Page 4 (Cont'd)

Observe that the projected output is now in terms of boxes of pencils. The first time this line was run, it was stated that the design goal was 54 pencils per minute. This translates to 6.75 boxes. Clearly, the design is nowhere near there yet.

When considering the logbook, be aware that TRES is still in terms of pencils only. No boxes are included there yet, since in this model, the boxes only are found in the output buffer, which is not included in TRES. As stated earlier, however, TRES does include those parts in delay between Machine 2 and Buffer 2.

```
          ACME LINE WITH SUPPLY

          SYSTEM LOG
          ----------


INT=      1.19  MACHINE  3 DOWN. DUE BACK UP AT     2. TRES=         0.
INT=      2.09  MACHINE  3 BACK UP. NEXT FAILURE AT    26
INT=      4.60  MACHINE  4 DOWN. DUE BACK UP AT    17. TRES=       245.
INT=      7.26  MACHINE  2 DOWN. DUE BACK UP AT    12. TRES=       490.
INT=      8.27  MACHINE  7 DOWN. DUE BACK UP AT     9. TRES=       572.
INT=      9.70  MACHINE  7 BACK UP. NEXT FAILURE AT   156
INT=     12.80  MACHINE  2 BACK UP. NEXT FAILURE AT    22
INT=     14.80  MACHINE  1 DOWN. DUE BACK UP AT    15. TRES=       846.
INT=     15.00  MACHINE  1 BACK UP. NEXT FAILURE AT    70
INT=     17.38  MACHINE  4 BACK UP. NEXT FAILURE AT   138
INT=     22.91  MACHINE  2 DOWN. DUE BACK UP AT    39. TRES=      1104.
INT=     26.97  MACHINE  3 DOWN. DUE BACK UP AT    36. TRES=       970.
```

Figure **49** – Example 3.5 – Logbook

3.6 Gaussian Failures

Once again, several changes will be made at one time to our model.

a. The specifications for our stamping machines will be revised. Even though the current design of two machines working at 33 PPM has the same expected output of a single machine working twice as fast at the same availability, clearly the rate of 33 is inadequate in the short run. So we will now design for a rate of 45, with no change in availability, in an attempt to keep product moving.

b. Since the required output from Machine 6 is equivalent to 54 PPM, its speed will have to be increased, now that it is coupled with an imperfect crimp supply. Dividing by the availability of the crimp supply (.9091) yields a new speed of 66 PPM.

c. The failure distribution from Machines 1, 2, and 7 will be changed to normal (Gaussian) to show the effect this has on the spread of failure times. Lacking other information, we arbitrarily set the sigma on MTBF to one fifth of the mean.

d. To reduce the burden on the early buffers somewhat, we will try running the model with only 77 parts of raw material per interval instead of the 82.5 we have used in the past.

e. In view of the occasionally loaded buffers experienced earlier, we will up our safety margin from 10% to 20%.

f. To get a better idea of how this line might work in the long run, we will exercise the model for 120 intervals, but not print out the machine handling table, as it would be too voluminous. Similarly, we will change the format of the periodic buffer status table to the standard form in order to get more utilization information. However, we will only print it every tenth interval.

```
000000000000000000000
ACME LINE WITH SUPPLY
006
TRIM      PAINT     STAMP     SUPPLY    CRIMP     PACKAGE
007
  1  1  1  0  1    82.50   36.00   7.200   4.000   0.000   0.010  1  0
  2  2  1  1  2    74.25   48.00   9.600   6.000   0.000   0.000  1  3
  3  3  1  2  3    45.00   40.00   0.000   4.000   0.000   0.000  0  0
  4  3  2  2  3    45.00   40.00   0.000   4.000   0.000   0.000  0  0
  5  4  1  3999    0.000   50.00   0.000   5.000   0.000   0.000  5  0
  6  5  1  3  4    66.00   45.00   0.000   5.000   0.000   0.000 -5  0
  7  6  1  4  5    64.00   50.00   10.00   6.000   0.000   0.000  7  8
005
  1   0.00000   535.000    1
  2   0.00000   430.000    1
  3   0.00000   360.000    1
  4   0.00000   390.000    1
  5   0.00000   0.00000    1
   77.000    420.000    0    0    0    0  1.00000
000000000000000000000
  120  10     0    0    0
000000000000000000000
```

Figure 50 — Example 3.6 — Input Deck

```
          ACME LINE WITH SUPPLY

                           INPUT  OUTPUT           FAILURE        REPAIR     NOMINAL  COMP   AUX   DEFECT
MACHINE  OPERATION(SEQ)   BUFFER  BUFFER   RATE   MEAN  SIGMA   MEAN  SIGMA  AVAIL.   TYPE  FIELD   RATE
-------  --------------   ------  ------   ----   ----  -----   ----  -----  -------  ----  -----  ------
   1     TRIM      ( 1)     0        1     82.5   36.0    7.2    4.0    0.0   .9000    1      0     .01000
   2     PAINT     ( 1)     1        2     74.3   48.0    9.6    6.0    0.0   .8889    1      3    0.00000
   3     STAMP     ( 1)     2        3     45.0   40.0    0.0    4.0    0.0   .9091    0      0    0.00000
   4     STAMP     ( 2)     2        3     45.0   40.0    0.0    4.0    0.0   .9091    0      0    0.00000
   5     SUPPLY    ( 1)     3      999      0.0   50.0    0.0    5.0    0.0   .9091    5      0    0.00000
   6     CRIMP     ( 1)     3        4     66.0   45.0    0.0    5.0    0.0   .9000   -5      0    0.00000
   7     PACKAGE   ( 1)     4        5     64.0   50.0   10.0    6.0    0.0   .8929    7      8    0.00000

GROUPED COMPONENTS
------------------

          GROUP    LAST ELEMENT   OUTPUT BUFFER
          -----    ------------   -------------
            1           6               4

TIMES OF FIRST FAILURE
----------------------
 (  1=    25)  ( 2=    41)  (  3=    8)  (  4=    39)  (  5=    33)  (  6=     5)  (  7=    42)

CONTROL POINT DATA
------------------

CONTROL   INITIAL    MAXIMUM    STATUS                   MAXIMUM  MAXIMUM
 POINT    CONTENTS   CAPACITY   PRINT   ENTRIES  EXITS    INPUT    OUTPUT
-------   --------   --------   ------  -------  -----   -------  -------
   1         0.         535.     YES       1       1      82.5     74.3
   2         0.         430.     YES       1       2      74.3     90.0
   3         0.         360.     YES       2       1      90.0     66.0
   4         0.         390.     YES       1       1      66.0     64.0
   5         0.     ********     YES       1       0       8.0      0.0   (OUTPUT POINT)

          RAW MATERIAL AVAILABLE=       77.00 PER INTERVAL
          SHIFT LENGTH=                420.00 INTERVALS
          LENGTH OF SIMULATION=        120    INTERVALS
          BASIC TIME INTERVAL=           1.00 MINUTES
--------------------------------------------------------------------------------------------------------
```

Figure 51 — Example 3.6 — Page 1

```
TOTAL PARTS HANDLED BY EACH MACHINE
-----------------------------------
 (  1=      6431.)  (  2=      5831.)  (  3=      2950.)  (  4=      2228.)  (  5=          0.)  (  6=      4818.)
 (  7=      4768.)

TOTAL RAW MATERIAL CONSUMED=      6431.
TOTAL MATERIAL IN-PROCESS=        1598.

AVAILABILITY DATA          (BY MACHINE )
-----------------

                            TOTAL      TOTAL      TOTAL       AVERAGE         S T O P   T I M E S
MACHINE  OPERATION(SEQ)   FAILURES   DOWN TIME   UP TIME   AVAILABILITY   DOWNSTREAM   UPSTREAM   TOTAL
-------  --------------   --------   ---------   -------   ------------   ----------   --------   -----
    1    TRIM       ( 1)      3          17        103         .8583           33           0        33
    2    PAINT      ( 1)      2          20        100         .8333           22           7        29
    3    STAMP      ( 1)      4           8        112         .9333           22          17        39
    4    STAMP      ( 2)      3          11        109         .9083           39          20        59
    5    SUPPLY     ( 1)      4          20        100         .8333            0           0         0
    6    CRIMP      ( 1)      6          12        108         .9000            0          13        13
    7    PACKAGE    ( 1)      2           5        115         .9583            0          33        33

TOTAL UNITS REJECTED=     66.

AVAILABILITY DATA         (BY CATEGORY)
-----------------
                     NO. OF      TOTAL      TOTAL      TOTAL       AVERAGE      AVERAGE
      CATEGORY      MACHINES   FAILURES   DOWN TIME   UP TIME   AVAILABILITY   STOP TIME
      --------      --------   --------   ---------   -------   ------------   ---------
( 1)  TRIM              1          3          17        103         .8583         33.0
( 2)  PAINT             1          2          20        100         .8333         29.0
( 3)  STAMP             2          7          19        221         .9208         49.0
( 4)  SUPPLY            1          4          20        100         .8333          0.0
( 5)  CRIMP             1          6          12        108         .9000         13.0
( 6)  PACKAGE           1          2           5        115         .9583         33.0

           24 TOTAL FAILURES REQUIRED        93 INTERVALS REPAIR TIME
          AVERAGE REPAIR TIME=     3.88

          SYSTEM WAS DEFECT-FREE FOR     57 INTERVALS OUT OF  120
          DEMAND FOR MAINTENANCE ACTION= .5250
```

Figure 52 — Example 3.6 — Page 2

All the changes discussed above can readily be accounted for in the input deck. Most of them, in fact, pertain to the Execute Card alone, to variables NSIMUL, NFREQ, NSPEC, and IREW2.

These changes are also reflected in the appropriate places on the first page of the output. Note especially the reasonable set of times to first failure. Only two machines had very early failures and they were among the machines still assigned an exponential failure distribution.

Since the individual machine handling table was turned off, the next page of output begins with the total parts handled by each machine. From this listing, it is possible to form an idea about line balance by comparing total parts handled by each station and the differences (delta) between these totals.

Station	1		2		3		4		5
Total	6431		5831		5178		4818		4768
Delta		600		653		360		50	

Station 3 represents the sum of Machines 3 and 4. Ideally, we would like to have these deltas approximately equal, representing a uniform decrease of parts flow through the line.

Note that Machine 5 (Supply) was down for 20 intervals. This automatically shut down its partner, Machine 6 (Crimp), which was down for 12 intervals itself. Some of this downtime undoubtedly overlapped, for we note that Machine 7 (Package) was starved for parts for a total of 33 intervals. Normally, it would see no parts for the first seven intervals of any run. Thus, if there were no overlap, its total upstream stoptime for this run would have been 39 intervals, rather than 33. In fact, we note that Machine 6 actually went down in the fifth interval, during the line's "warm up" period, further complicating the matter.

There were a total of 66 pencils rejected by Machine 1, slightly higher than 1% of its total output.

One final consequence of the change from exponential to normal failure distributions on several machines is a slightly lower demand for maintenance action (52.5%) compared to the last run (73.3%). This is a direct result of seeing most failures occurring later in the run. On the strength, however, of only a 120-interval run, no real significance should be placed upon this value yet.

```
                 ACME LINE WITH SUPPLY

                 PERIODIC BUFFER STATUS
                 ----------------------

INTERVAL         1        2        3        4        5
-------------------------------------------------------
   10           94.      74.     134.      70.      21.
                85.      45.      56.      24.       4.
             100/ 0   100/ 0    60/40   100/ 0   100/ 0
               0/ 0     0/ 0     0/ 0     0/ 0     0/ 0
                94.      74.     134.      70.      21.
-------------------------------------------------------
   20          113.      74.     216.      90.     101.
               104.      90.     164.      81.      65.
             100/ 0    90/10    30/50   100/ 0   100/ 0
               0/ 0     0/ 0    20/ 0     0/ 0     0/ 0
               113.     126.     216.      90.     101.
-------------------------------------------------------
   30           76.      62.     360.      72.     176.
                77.      74.     302.      75.     143.
              99/ 0    93/ 6    19/33    96/ 3    99/ 0
               0/ 0     0/ 0    26/19     0/ 0     0/ 0
               121.      85.     360.      98.     176.
-------------------------------------------------------
   40           96.     114.     360.      76.     241.
                87.      70.     343.      59.     210.
             100/ 0    90/10    15/25    97/ 2   100/ 0
               0/ 0     0/ 0    20/40     0/ 0     0/ 0
                96.     115.     360.      76.     241.
-------------------------------------------------------
   50          535.       0.       0.     240.     288.
               392.      47.     201.     211.     261.
              82/ 4    88/12    18/22    82/ 4   100/ 0
               2/12     0/ 0    18/42    14/ 0     0/ 0
               535.     179.     360.     292.     288.
-------------------------------------------------------
   60          533.      74.      83.      66.     323.
               535.      26.      20.      44.     314.
              68/ 3    89/ 9    31/18    81/ 6    99/ 0
               1/26     0/ 0    14/34    11/ 0     0/ 0
               535.      74.      83.     176.     323.
-------------------------------------------------------
   70            0.     272.     114.      74.     386.
               165.     151.     148.      55.     358.
              65/ 5    82/14    27/27    84/ 5    99/ 0
               4/24     2/ 0    15/29     9/ 0     0/ 0
               459.     272.     219.      74.     386.
-------------------------------------------------------
INTERVAL         1        2        3        4        5
```

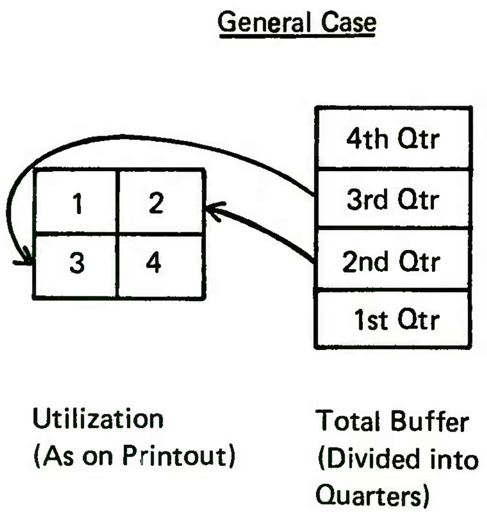
Figure 53 — Example 3.6 — Page 3

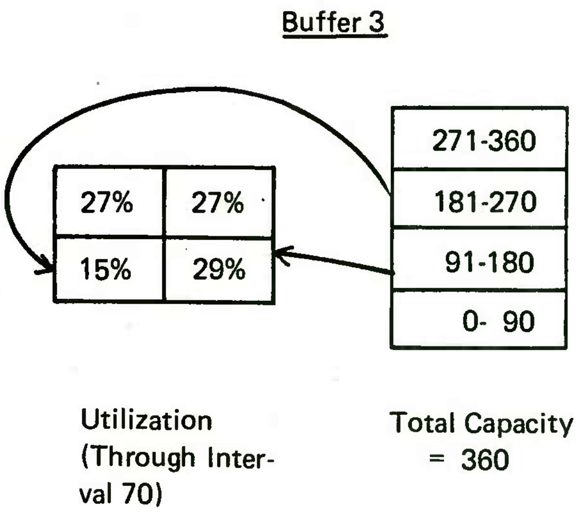
In fact, no conclusions should be reasonably drawn about any aspect of this line on the basis of the runs we have made so far. These runs and the discussion that accompanies them are presented solely to illustrate the mechanics of GENMOD and the utility of its various outputs. They are designed to alert the user to key phrases to look for and how to interpret what might otherwise appear to be contradictory information. While this manual tries to present a complete picture of GENMOD, it is not intended to be a simulation manual. It is expected that the reader already has a feel for the proper application of simulation techniques and that he is simply looking for another tool.

Some explanation must now be given of the new form of the periodic buffer status table. First of all, it is printed every tenth interval rather than every single interval, as specified on the EXECUTE card (variable NFREQ). Secondly, it consists of five lines of information per printout, rather than just one (variable IREW2). For a detailed account of the meaning of these lines, consider the block of information for Buffer 3 as the end of Interval 70:

114.	=	Current contents of the buffer	
148.	=	Average contents since last printout	
27/27	=	% Utilization of quarters 1 and 2	(See Figure 54)
15/29	=	% Utilization of quarters 3 and 4	
219.	=	Maximum contents since last printout	

In other words, after 70 intervals, Buffer 3 is seen to contain 114 pencils. During the preceding ten intervals, i.e. since Interval 60, its contents have averaged 148, reaching a high of 219. For the total elapsed time of 70 intervals, Quarter 1 (0-90 parts) was sufficient 27% of the time, Quarter 2 (91-180 parts) 27%, and so on. (Note that these percentages are truncated integers and rarely sum to 100%.)


General Case
1
2
3
4
4th Qtr
3rd Qtr
2nd Qtr
1st Qtr
Utilization (As on Printout)
Total Buffer (Divided into Quarters)



Each buffer is mentally divided by GENMOD into quarters. An account is kept of how often the current contents of the buffer (at the end of an interval) fell into one of these quarters. The printout expresses the utilization of these quarters as the percentage of current elapsed time at that quarter or higher was needed to hold parts. (See text)

Figure 54 – Buffer Quarter Utilization

This page intentionally left blank.

The purpose of these quarterly utilization percentages is to help size buffers by indicating how often a buffer with a different capacity would have sufficed. If, at the end of a run, most of the utilization is in the 1st and 2nd quarters, then probably a buffer half as large as designed could be tried. Likewise, if most of the utilization is in the upper quadrants, than a larger buffer should be considered. The following are typical examples of utilization patterns and resultant actions that might be considered.

6/93	=	Reduce buffer capacity by 50%
0/ 0		
25/15	=	Might consider raising capacity slightly
20/40		
50/ 0	=	Examine operating characteristics of stations entering and exiting
0/50		this buffer.
6/81	=	Reduce buffer by perhaps 1/3
11/ 0		
11/ 0	=	Increase buffer by about 50%
6/81		
30/30	=	Leave as is
13/26		
5/ 4	=	Reduce slightly and examine characteristics of exiting station
90/ 0		

With regard to the buffer status table for this particular run, the reader is invited to trace some of the values himself. Note that the first three buffers all hit their maximum capacity at one time or another, with Buffer 3 becoming full after only 30 intervals. This is reflective of the relatively poor performance of the Supply/Crimp station. The expected output is slightly higher than previously. To help reconstruct parts flow, the entire logbook is presented.

```
          ACME LINE WITH SUPPLY

          PERIODIC BUFFER STATUS
          ----------------------

INTERVAL           1        2        3        4        5
-------------------------------------------------------
    80            90.      74.      91.      68.     455.
                  68.      66.      92.      66.     428.
                70/ 5    82/15    30/30    86/ 5   100/ 0
                 3/21     2/ 0    13/26     8/ 0     0/ 0
                  90.     207.     174.      86.     455.
-------------------------------------------------------
    90           109.      86.     360.       0.     522.
                 101.      75.     175.      55.     497.
                73/ 4    84/13    26/34    87/ 4    99/ 0
                 3/18     2/ 0    13/25     7/ 0     0/ 0
                 109.      86.     360.      82.     522.
-------------------------------------------------------
  100            461.     430.     360.       0.     530.
                 212.     336.     360.       7.     529.
                71/ 6    77/13    24/31    89/ 4   100/ 0
                 5/18     4/ 6    12/33     7/ 0     0/ 0
                 461.     430.     360.      66.     530.
-------------------------------------------------------
  110            535.     430.     360.       0.     554.
                 535.     430.     360.      21.     537.
                64/ 5    69/11    21/28    89/ 3    99/ 0
                 4/25     3/14    10/39     6/ 0     0/ 0
                 535.     430.     360.      68.     554.
-------------------------------------------------------
  120            535.     430.     360.      51.     596.
                 535.     430.     360.      41.     567.
                59/ 4    64/10    19/25    90/ 3    99/ 0
                 4/31     3/21     9/44     5/ 0     0/ 0
                 535.     430.     360.      74.     596.

MAXIMUM
MATERIAL
HANDLED-         535.     430.     360.     292.     596.


PROJECTED AVERAGE OUTPUTS
-------------------------
        PER BASIC INTERVAL-              4.97 (    4.97 PER MINUTE)
        PER 420.-INTERVAL SHIFT-     2086.
        PER 3/8/5 WEEK        -     31288.
        PER 63-SHIFT MONTH    -    131409.
```

Figure 55 — Example 3.6 — Page 3 (Cont'd)

```
                ACME LINE WITH SUPPLY

                SYSTEM LOG
                ----------

INT=      5.27   MACHINE  6 DOWN. DUE BACK UP AT    7. TRES=       305.
INT=      7.59   MACHINE  6 BACK UP. NEXT FAILURE AT    25
INT=      8.91   MACHINE  3 DOWN. DUE BACK UP AT    9. TRES=       533.
INT=      9.99   MACHINE  3 BACK UP. NEXT FAILURE AT    13
INT=     13.19   MACHINE  3 DOWN. DUE BACK UP AT   15. TRES=       619.
INT=     15.85   MACHINE  3 BACK UP. NEXT FAILURE AT    41
INT=     25.92   MACHINE  1 DOWN. DUE BACK UP AT   29. TRES=       766.
INT=     25.55   MACHINE  6 DOWN. DUE BACK UP AT   26. TRES=       766.
INT=     26.24   MACHINE  6 BACK UP. NEXT FAILURE AT    63
INT=     29.78   MACHINE  1 BACK UP. NEXT FAILURE AT    60
INT=     33.78   MACHINE  5 DOWN. DUE BACK UP AT   35. TRES=       655.
INT=     35.33   MACHINE  5 BACK UP. NEXT FAILURE AT    41
INT=     39.25   MACHINE  4 DOWN. DUE BACK UP AT   40. TRES=       844.
INT=     40.87   MACHINE  4 BACK UP. NEXT FAILURE AT    42
INT=     41.88   MACHINE  2 DOWN. DUE BACK UP AT   57. TRES=       869.
INT=     41.27   MACHINE  3 DOWN. DUE BACK UP AT   42. TRES=       869.
INT=     41.28   MACHINE  5 DOWN. DUE BACK UP AT   42. TRES=       869.
INT=     42.67   MACHINE  3 BACK UP. NEXT FAILURE AT    48
INT=     42.20   MACHINE  4 DOWN. DUE BACK UP AT   45. TRES=       881.
INT=     42.22   MACHINE  5 BACK UP. NEXT FAILURE AT    88
INT=     42.13   MACHINE  7 DOWN. DUE BACK UP AT   46. TRES=       881.
INT=     45.90   MACHINE  4 BACK UP. NEXT FAILURE AT    64
INT=     46.84   MACHINE  7 BACK UP. NEXT FAILURE AT   101
INT=     48.56   MACHINE  3 DOWN. DUE BACK UP AT   52. TRES=      1189.
INT=     52.68   MACHINE  3 BACK UP. NEXT FAILURE AT   289
INT=     57.54   MACHINE  2 BACK UP. NEXT FAILURE AT    98
INT=     60.96   MACHINE  1 DOWN. DUE BACK UP AT   72. TRES=       946.
INT=     63.78   MACHINE  6 DOWN. DUE BACK UP AT   65. TRES=       850.
INT=     64.40   MACHINE  4 DOWN. DUE BACK UP AT   71. TRES=       786.
INT=     65.00   MACHINE  6 BACK UP. NEXT FAILURE AT   102
INT=     71.75   MACHINE  4 BACK UP. NEXT FAILURE AT   260
INT=     72.22   MACHINE  1 BACK UP. NEXT FAILURE AT   104
INT=     88.87   MACHINE  5 DOWN. DUE BACK UP AT   91. TRES=       631.
INT=     91.58   MACHINE  5 BACK UP. NEXT FAILURE AT    92
INT=     92.17   MACHINE  5 DOWN. DUE BACK UP AT  106. TRES=       854.
INT=     98.82   MACHINE  2 DOWN. DUE BACK UP AT  102. TRES=      1245.
INT=    101.53   MACHINE  7 DOWN. DUE BACK UP AT  102. TRES=      1474.
INT=    102.55   MACHINE  2 BACK UP. NEXT FAILURE AT   154
INT=    102.43   MACHINE  6 DOWN. DUE BACK UP AT  103. TRES=      1547.
INT=    102.97   MACHINE  7 BACK UP. NEXT FAILURE AT   148
INT=    103.17   MACHINE  6 BACK UP. NEXT FAILURE AT   108
INT=    104.79   MACHINE  1 DOWN. DUE BACK UP AT  105. TRES=      1548.
INT=    105.36   MACHINE  1 BACK UP. NEXT FAILURE AT   133
INT=    106.33   MACHINE  5 BACK UP. NEXT FAILURE AT   214
INT=    108.97   MACHINE  6 DOWN. DUE BACK UP AT  114. TRES=      1615.
INT=    114.19   MACHINE  6 BACK UP. NEXT FAILURE AT   120
INT=    120.61   MACHINE  6 DOWN. DUE BACK UP AT  121. TRES=      1621.
```

Figure 56 — Example 3.6 — Logbook

3.7 Additional Topics

At this point we shall leave the Acme Pencil Line to discuss in greater detail some topics presented earlier. Since the main purpose of these discussions is to show the proper use of various GENMOD features, we will be concerned mostly with the composition of the input deck, with or without actual sample runs.

3.7.1 Seeding Buffers

All the sample runs for our pencil line were made with the initial contents of all buffers being zero. This was done chiefly to make it easier to follow the product flow through the model.

In actual use, however, starting with empty buffers would normally be done for one of two reasons:

a. to debug a newly constructed or modified model, or

b. to simulate running from a dead start in order to estimate the time needed to reach steady state production.

The first case is similar to what we have done. Often the complexity of a model requires an interval-by-interval examination of a run to determine if everything is modeled and coded correctly. Good judgement must be exercised here, of course, since such a study can produce voluminous output that might prove tedious to examine thoroughly. The situation can be further complicated if MTBF's are of such a magnitude that failures do not occur until rather late in the run.

Simulating to determine steady state time is analogous to turning the line on for the first time bright and early Monday morning and waiting to see how long it takes for everything to run smoothly. This point is usually categorized by a leveling off of buffer contents and average production (output) rate. Normally, this point won't be reached until all machines with "reasonable" MTBF's have had a chance to fail at least once.

Under most circumstances, however, a production line will not consistently start running with empty buffers. Except for lines containing perishable or sensitive material which must be removed from all buffers at the end of a production run, it is not unusual to leave all in-process material right where it is at the end of a run. Thus, when the line is restarted, most buffers already have material in them and all machines can begin working immediately, rather than having to wait for parts to arrive. This is the usual simulation problem to be investigated. On a typical day, with buffers containing a typical remainder from the preceding day, what can we expect as a typical output?

The question of how much to seed each buffer with can be answered in several ways, and the user is completely free to input whatever values he chooses. One approach is to make several runs of sufficient duration with empty buffers and to note the average contents of all buffers, as printed on the last page of a standard output. The average of these averages will then be a good indicator of where to start.

Changing the seed for any buffer is very easy. Figure 57 shows the input deck for our last sample case, with the buffer seeds modified to reflect the average contents seen on that run.

```
0000000000000000000000
ACME LINE WITH SUPPLY
006
TRIM        PAINT      STAMP      SUPPLY     CRIMP      PACKAGE
007
  1  1  1  0  1   82.50    36.00    7.200    4.000    0.000    0.010   1   0
  2  2  1  1  2   74.25    48.00    9.600    6.000    0.000    0.000   1   3
  3  3  1  2  3   45.00    40.00    0.000    4.000    0.000    0.000   0   0
  4  3  2  2  3   45.00    40.00    0.000    4.000    0.000    0.000   0   0
  5  4  1  3999    0.000    50.00    0.000    5.000    0.000    0.000   0   0
  6  5  1  3  4   66.00    45.00    0.000    5.000    0.000    0.000   5   0
  7  6  1  4  5   64.00    50.00    10.00    6.000    0.000    0.000  -5   0
005                                                                   7   8
  1    241.000    535.000    1
  2    153.000    430.000    1
  3    215.000    360.000    1
  4    62.0000    390.000    1
  5    0.00000    0.00000    1
   77.0000    420.000     0     0     0     0    1.00000
0000000000000000000000
  120    10     0     0     0
0000000000000000000000
```

Figure 57 — Input Deck for Seeding Buffers

Note that the output buffer (Buffer 5) is not seeded with any material. Doing so would be equivalent to saying that much material had already been processed, even before the first machine is turned on. Only under special circumstances would such a trick be meaningful, namely to try to continue a particular simulation run, something that will not be discussed at this time.

If the need arises, it is very easy to isolate a particular machine, or station, or portion of a model to study its behaviour. This can be done by starting all buffers empty except the one feeding the station in question and then running the model with no raw material. Thus, no new material will be introduced to the line and the net result will be watching the station under investigation process everything in its input buffer and pass it downstream.

Many interesting studies can be accomplished through careful assignment of buffer seed values.

3.7.2 Uncapped Buffers

In our pencil line examples, we made several adjustments to the theoretical maximum of all the buffers. This was done to illustrate how a known "designed" capacity can easily be incorporated into the model and then changed if need be.

To be sure, every line should be designed with some idea of buffer requirements borne in mind. However, there are many times when it might be desirable to not put a specific cap on a buffer. For example, buffers for a particular product may not be very expensive and therefore their total size may not be of great importance.

Also, there may be situations too complex to realistically estimate the exact size needed. In such cases, though, it may be known that a buffer larger than some particular size might be prohibitive. Therefore, all buffers might be given this maximum cap to insure that nothing greater is ever called for.

Finally, we may be interested not so much in telling the model what to use, but rather in having the model tell us what we need. Just as the output buffer is always given an infinite capacity (since anything else would generally be meaningless), it is perfectly acceptable to give all buffers an unlimited capacity. We would then, in effect, be allowing the model to run without restrictions and to allow the buffers to attain whatever level is necessary to keep parts moving. After making a few such runs of reasonable duration it would then be possible to make some judgement as to whether the resultant buffer demands are acceptable or whether the line must be redesigned.

An unlimited capacity can be given to any buffer, simply by inputting a zero, as for the output buffer.

3.7.3 Renumbering Machines

One of the first conventions about GENMOD ever mentioned was that machines are numbered from top to bottom, left to right, starting at one. If this were strictly enforced, it would be difficult to combine sub-models into a larger one, as many machines would have to be renumbered, requiring many cards to be repunched.

Strictly speaking, all machines within GENMOD must be numbered from one. However, by means of another option, the machines on the flow chart and the cards can be numbered beginning anywhere. The numbers will automatically be adjusted as the cards are read. To do this, on the data card that contains the count of machines in the model, simply append another three-digit integer containing the number of the first machine in the model. If left blank, as in all examples shown thus far, this number is assumed to be one.

This arrangement allows for independent testing of small portions of large models and also allows for future expansion of existing models. A couple of examples will clarify this option.

Suppose the following three sub-lines are ultimately to be constructed serially.

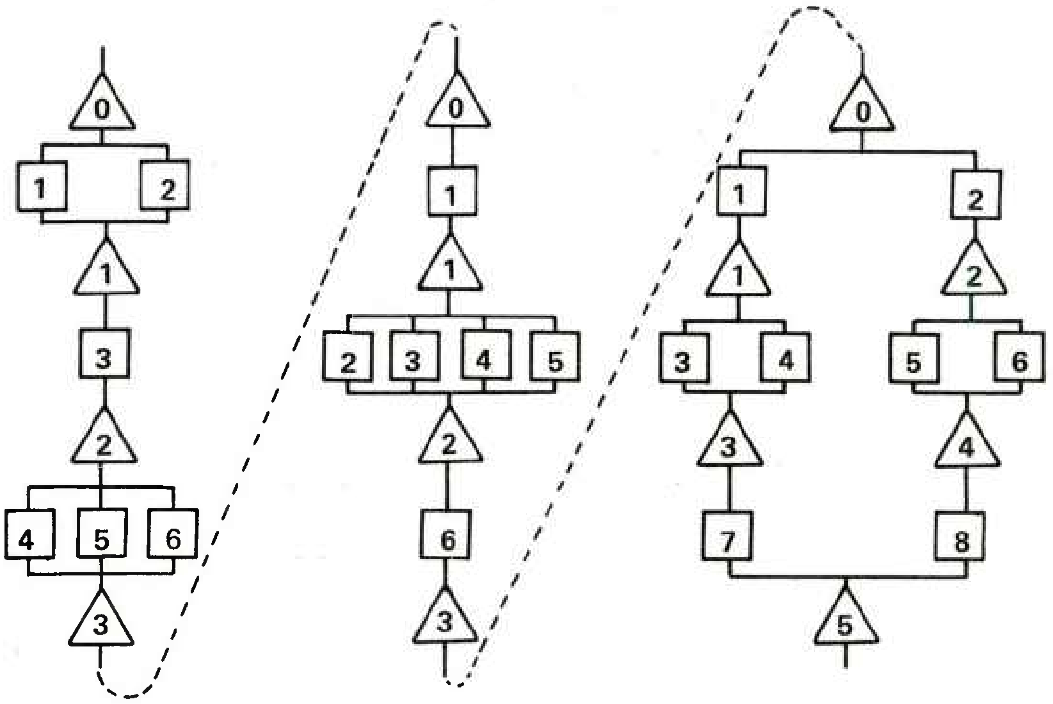


Figure 58 – Three Serial Sub-Lines

The machines and buffers are numbered in the figure as if they were part of three unrelated entities. How can three input decks be punched now so that at a later date a combined model can be run with a minimum of repunching? The first sub-line can be punched exactly as is, with six machines and three buffers. The second sub-line really contains Machines 7 through 12 and Buffers 4 through 6. The last sub-line has Machines 13 through 20 and Buffers 7 through 11. Figure 59 shows these same sub-lines, with the machines and buffers numbered as they really would be on the complete line.

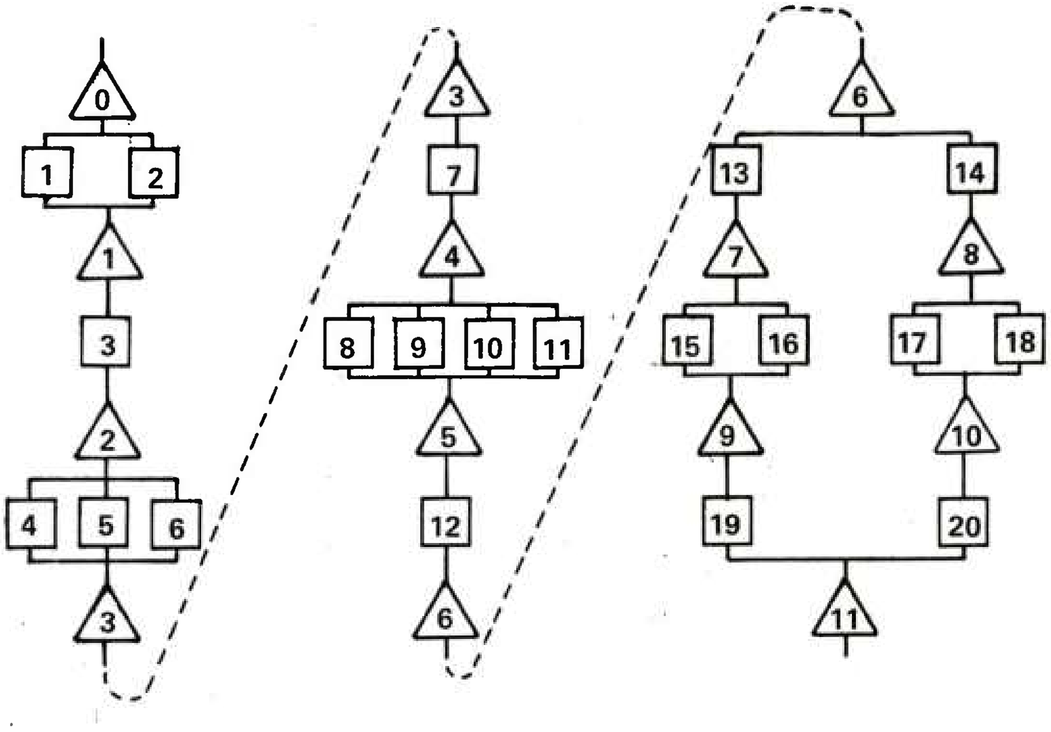


Figure 59 — Re-Numbered Sub-Lines

The second sub-section can now also be punched directly with a few minor adjustments:

a. The number of categories should be six, instead of three, since eventually the category titles used in the first sub-line will be incorporated into the complete model. The machines here, then, will be coded as belonging to Categories 4, 5, and 6.

b. The number of machines should be coded as being six, beginning at seven.

c. Buffer 3 must temporarily be renumbered as Buffer 0, as all GENMOD models must have a Buffer 0. However, all other buffers can be left as they are. Machine 7 will be coded as going from Buffer 0 to Buffer 4, which is perfectly acceptable. Since we go up to Buffer 6, we must state that we have six buffers and must provide six buffer cards. Since this portion of the model really only uses Buffers 4, 5 and 6, it would be alright to enter dummy information for Buffers 1, 2, and 3. But, these buffers really exist. So, why not use the correct information?

d. When run by itself, this section must be provided with an appropriate value for raw material.

Corresponding adjustments should be made when running the third section by itself, also.

Figure 60 shows a proper input deck for the second section, with all irrelevant data X'd out.

```
0000000000000000000000
NUMBERING EXAMPLE - SECTION 2
006
STEP1      STEP2      STEP3      STEP4      STEP5      STEP6
006  7
  7  4  1  0  4   XX.XX   XX.XX   XX.XX   XX.XX   XX.XX   .XXXX  X  X
  8  5  1  4  5   XX.XX   XX.XX   XX.XX   XX.XX   XX.XX   .XXXX  X  X
  9  5  2  4  5   XX.XX   XX.XX   XX.XX   XX.XX   XX.XX   .XXXX  X  X
 10  5  3  4  5   XX.XX   XX.XX   XX.XX   XX.XX   XX.XX   .XXXX  X  X
 11  5  4  4  5   XX.XX   XX.XX   XX.XX   XX.XX   XX.XX   .XXXX  X  X
 12  6  1  5  6   XX.XX   XX.XX   XX.XX   XX.XX   XX.XX   .XXXX  X  X
006
  1    XXX.XXX    XXX.XXX    X
  2    XXX.XXX    XXX.XXX    X
  3    XXX.XXX    XXX.XXX    X
  4    XXX.XXX    XXX.XXX    X
  5    XXX.XXX    XXX.XXX    X
  6    XXX.XXX    0.00000    X
   XXX.XXX    XXX.XXX    X    X    X    X   XX.XXXX
0000000000000000000000
```

Figure 60 — Sub-Line Input Deck — Section 2

In order to combine this model with the first section, all that is necessary is to a) merge the two sets of machine cards together and change the number of machines to 12, b) change the input buffer for Machine 7 from 0 to 3, and c) make sure that all buffers have the correct maximum capacity, especially Buffer 3, which served as the output buffer for the first section. If the buffer cards for the second section were punched correctly originally, then no changes are necessary at all. Simply throw out the buffer cards for the first section.

When ultimately adding the third section, similar modifications can be made. In general, if some planning is used beforehand, sub-models can be combined with little difficulty.

Suppose now that a finalized design is not yet available for the first half of a line, but we can proceed with a simulation of the second portion. We ultimately want to combine the two, doing as little repunching as necessary. We do know, though, that the first portion will probably have about ten machines, using three categories, and five buffers. Figure 61 shows the extent of our information at present.

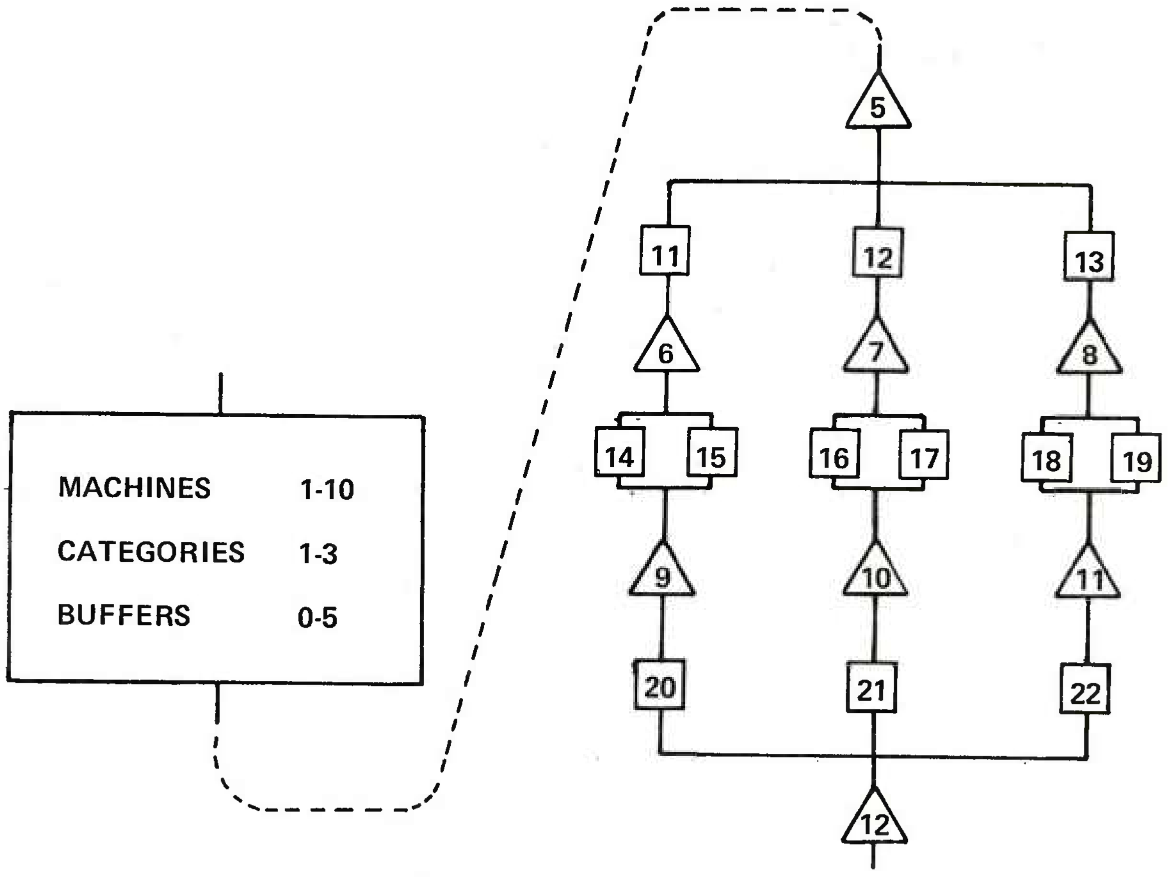


Figure 61 — Future Line Expansion

The second portion can be punched directly now, as in the preceding example, beginning with Machine 11, allowing for six categories and using twelve buffers. We can use dummy names for the first three categories and dummy values for the first five buffers. There is never any penalty in GENMOD for specifying more information than is necessary. If a category is defined but never used, GENMOD will make note of that fact. Also, if a buffer is defined but never used, it will be ignored. The converse, however, is not true. Never assign a category number without defining it or route material to a buffer that doesn't exist.

Lastly, what if the first section of this line, when designed, really only had eight machines and four buffers. Has any damage been done by assuming otherwise in the first place? No, because extra or unused machine numbers can be eliminated by using Code 99 and extra buffers are ignored. No difficulty should be encountered at all by combining both portions of this line at a later date.

The main things to remember when using this renumbering option, whether for assessing sub-models or allowing for future expansion, are:

a. Don't be conservative. It is much easier to turn off an unused machine than it is to try to squeeze in an extra one. Likewise with ignoring unused categories or buffers.

b. When punching the actual cards, do not skip any numbers. While you have the option of starting your sequence numbers anywhere you wish, once that sequence begins all machines must then be numbered consecutively. If a particular machine number really does not exist in your model, you must punch a card anyway, using Type Code 99. Remember also that the count of machines in your model must include all Code 99's.

c. The lead buffer in a sub-model must temporarily be renumbered as Buffer 0 and the amount of raw material must be adjusted accordingly.

d. If the buffers in a particular sub-model do not start from one, additional buffer cards must be placed in the deck, even with dummy parameters if need be.

3.7.4 Recycle Loops

Up to now we have thought of all product in GENMOD as moving in one direction, from top to bottom, in the model. There is no reason, however, why product can't go back upstream if no real damage is done by possibly mixing in one buffer parts in different stages of production.

A frequent need for routing material back upstream exists on chemical lines, where process solutions are often recycled and brought back onto the line at various places. This recycled material may or may not be part of the finished product. But there may be the desire to model such recycle loops.

On hardware lines, different situations may exist. A common need for recycling involves "semi-defective" material which can be reworked and sent to the appropriate buffer for reentry on the line. Our long pencils fit this category. They can be sent back and retrimmed. But the short ones must be scrapped.

Another recycle loop might concern the distribution of certain fixtures employed by the manufacturing process, or with re-routing of boxes or trays, etc.

In all cases, though, it is imperative that a decision be made as to whether the material being recycled is or is not part of the true finished product. Recall that the mainstream of GENMOD is considered to consist solely of the ultimate product in various stages of manufacture. Be careful that material is at all times routed to buffers containing like material. This statement is true of all GENMOD material movements, not just recycle loops.

Since GENMOD is machine-oriented, it is very easy to route material to any buffer whatsoever (except for Buffer 0) simply by indicating the correct buffer number for OUT on the data card. GENMOD will automatically set up the routing, no matter how unorthodox, and will attempt to move parts in that direction. Thus, simply sending material back upstream can be accomplished with no coding tricks at all.

To move a specific amount of material back up, though, is a little more complex and is usually best done by means of proportional splitters. Consider Figure 62, which shows 10% of the output from Machine 2 going back to Buffer 1, via Machine 4. Here, Machine 2 is doing real work and passing the material to Buffer 2, from which it is then distributed by a pair of perfect proportional splitters. Machine 3 sends 90% downstream to Buffer 3, while the other 10% goes back upstream. It is not possible to have Machine 2 distribute the material itself for two reasons:

a. <u>no</u> GENMOD machine can feed more than <u>one</u> buffer, and

b. a proportional split requires a perfect dummy machine, which would detract from whatever significance the machine had in the first place.

3.7.5 <u>Inappropriate BTI Choices</u>

Very often a new user of GENMOD will get so involved in working out the intracies of various building blocks that he winds up losing sight of the information he wanted in the first place. Blocks will be incorporated for which data cannot be obtained. Assumptions are made which yield no new information. Restrictions are imposed which prevent meaningful answers. But, most often, extreme detail at one end of the model will be negated by gross generalizations at the other.

One of the more troublesome areas, though, involves the choice of the basic time interval. To be sure, the <u>ultimate</u> accuracy of a GENMOD model is a function of the BTI. But, how can we determine how much damage (if any) is done by using something other than the "ideal" BIT? Do we gain anything by changing, say, from one minute to ten seconds? Do we lose anything by switching from one minute to one hour?

As stated much earlier, the BIT is analagous to the duration between frames of a motion picture of the line. We gain nothing if the BTI is so small in relation to the action of

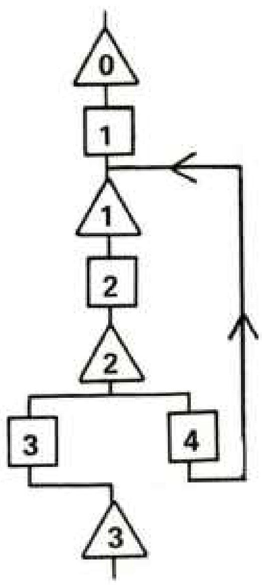


Figure 62 – Recycle Loop

the process that each frame is essentially the same as the previous one. Similarly, we have lost too much if the BTI is so long that everything on the film is a blur.

Usually the nature of the process and the machines will indicate an appropriate BTI. High speed, small piece manufacturing processes have a lot of action in short periods of time and usually need a short BTI, e.g., one minute. Slower lines, especially batch chemical processes, can use longer times, e.g. 30 minutes. Machines with long MTBF's may never be seen to fail within tolerably long runs if the BTI is too short. Yet, too much accuracy is lost if a long BTI is used when very short MTBF's or MTTR's are involved (such that scheduled events less than one interval long result).

Many aspects of a GENMOD simulation are directly effected by the BTI. The following discussion covers some of these points and indicates what troubles might result from using a grossly inappropriate basic time interval.

a. <u>Buffer Size</u> – At first it would not seem that the choice of a BTI could influence buffer sizes. However, remember that GENMOD only moves parts <u>once</u> each interval, namely at the end, and that <u>all</u> parts scheduled for movement are moved at one time. Thus, <u>the output buffer for any station must be at least as large as the maximum parts that can be moved by that station in a single interval</u>. For example, if a machine handles parts at ten per minute and its output buffer holds 100 parts, then choosing a BTI of, say, 15 minutes will never work, as no more than 100 parts will ever be moved at one time.

To a certain extent, the relationship between BTI and buffer size also has a bearing on stop times. The number of intervals required to fill a blocked buffer is obviously related to how much material is moved each interval. If a BTI is too large, the buffer will fill up rather quickly, in terms of intervals, thus shortening the line's reaction time.

Under no circumstances, though,should the BTI be allowed to influence buffer sizing. Buffers are real and have definite size that remains unchanged by what is going on elsewhere. If anything must be adjusted, it should be the basic time interval.

b. <u>In-Process Material</u> — The outputs presented thus far show quite clearly that a lot of time in GENMOD is spent putting parts in buffers, because no machine can work until there is something in its input buffer. Once a buffer receives parts, though, in a steady state condition the number of parts in that buffer remains constant. For example, Cells 1, 2, and 3 shown in Table 6 have a strong tendency to remain fixed at 74 parts, because an identical amount of material is put in and taken out each interval. The level to which buffers in a steady state will tend is related to the amount of material initially deposited in them by the entering machines. This, of course, is dependent upon the BTI. Quite simply, models with long BTI's will have larger amounts of material in-process than ones with short BTI's. Since this material value (TRES) is often used to estimate inventory costs, it is obviously wise to study its sensitivity to your choice of BTI.

c. <u>Residence Time</u> — It has been stated many times that residence time is the equivalent sum of real time needed to step through a GENMOD model, including delays and multi-interval operations. A model requiring 17 steps, for example, implies a residence time of 17 minutes if BTI = 1.0 and nearly three hours if BTI = 10.0 and so on. Just how critical is the difference?

Whenever a model is started with empty buffers, it takes a fixed number of intervals before any output is produced. This can be considered as lost production time. This time will be lost again whenever the lead station goes down long enough to bring the buffers back down to nearly empty. Surely it makes a difference, if we say we lose three hours when we really only lose 17 minutes, and vice versa.

Similar losses of production time occur whenever there is a failure of any kind. The length of time in <u>intervals</u> that it takes to recover is a matter of mechanics. The <u>real</u> equivalent of this time is a matter of BTI.

If a model is always started with seeded buffers, output will begin almost immediately and it may not be apparent that residence time means anything. The user may not even take into account the effect of lost production time. Also, the observed output may be confused with true output (i.e. after initial seeding is discounted).

It is always best if BTI can be approximately equal to total true residence time divided by the number of process steps , in intervals. This will allow for the greatest margin of error.

d. Event Times — Much has already been said about failure and repair times that must be rounded up to one interval. Whenever such events occur we lose the accuracy of seeing the true failure or repair time. If this only happens a few times per run, no damage is done. But, if a large proportion of event times are always rounded upward, the accumulated errors will be meaningful.

Occasionally the range of MTBF's or MTTR's for a line is such that when BTI is chosen so that no event has a mean time of less than, say, five intervals, then others wind up with a time of 5,000 or so. If this is really the case, nothing much can be done. The short BTI must be retained.

However, in certain circumstances it may be possible to redefine the meaning of a failure, such that the short MTBF's can be revised upward, with no loss of accuracy. In particular, minor adjustments to equipment that occur every few minutes and last but a few seconds need not be considered distinct failures. It is perfectly legitimate to simply degrade the rated speed of a machine to reflect that, say, 2% of production time is employed in adjustments. After all, if such adjustments are a designed feature of the machine, the speed should take them into account and they should not be called failures. Such common sense evaluations often will lead to more straight-forward models.

e. Timing Compatibility — Whenever a model is constructed, it should be done so with one single BTI in mind, e.g. one minute. All values can be pencilled in based on this time and changed just before punching if need be. (In fact, pre-processors exist for GENMOD which will take a data deck punched with one BTI and convert the whole thing to a different BTI.)

When changing from one BTI to another, be sure to adjust all time-related functions: MTBF, MTTR, delay time, speed, batch size, shift length, maintenance frequency, etc.

Certain things, though, do not get changed, such as buffer capacities, proportional splits, defect rates, packaging ratios, etc.

f. Length of Simulation — Clearly, the BTI influences the length of real time that is simulated on any run. A GENMOD run of 5000 intervals can represent a true production of anything from 21 hours with a BTI of 15 seconds to ten months on a 3/8/5 basis with a BTI of one hour.

Exactly how long to run a particular model depends upon what information is being sought, how complex the model is, and what the range of MTBF's is. As a rough rule, it would be wise to simulate for from two to three times the longest MTBF of a major machine. Just because an individual machine may have an MTBF of 2000 or so should not be the overriding consideration. If that machine has little impact on production or is quickly repaired, it can often be disregarded when deciding upon how long to simulate.

While we are usually interested in simulating at least until a steady state has been reached, we do not want to go beyond reason. A common mistake is to exercise a model for a longer period than the line is intended to exist or beyond a point when we are reasonably sure that major modifications would have been made. Running GENMOD for so long would probably yield more meaningful results if the BTI were shortened.

Since GENMOD attaches no significance to "real" time, only to a set number of intervals, one model of 5000 intervals is no different from any other of like duration (disregarding size). The reader is invited to consider the following problem. If a particular model is run for 5000 intervals with a BTI of 1.0 and then for 5000 intervals at 60.0 (with all appropriate changes being made properly), which run, if either, reflects more "confidence"?

3.7.6 Inspection Equipment

There are several ways to model inspection procedures, depending upon the amount of information sought. All inspections, though, must be considered as continuous with no feedback logic involved. That is, parts are inspected according to some definite rule that does not change with the detection of defects. Switching from a sampling mode to a screening mode, for example, cannot easily be incorporated into a model. GENMOD was not designed to provide that sort of detail. Yet, the net effect of such procedures in the long run can be studied with GENMOD.

The simplest form of inspection to model is 100% inspection, either by a manufacturing machine or by a separate piece of equipment. Simply place a non-zero value in the DEF field of the data card and approximately that much of all processed material passing through the machine will be discarded as being defective. Just do not try to make the machine perfect, i.e. it must have a non-zero MTTR. Also, the individual defects cannot be accounted for, as they are not placed in any particular buffer.

The next most complex form of inspection involves sampled destructive inspection, with or without defect accountability. Here, a fixed proportion of material is removed from the line and never brought back. To accomplish this, we must think of the inspection as separate from the processing machine and, furthermore, we must insert on the line a pair of proportional splitters, which can be thought of as doing the sampling. The arrangement would be very similar to that shown earlier in Figure 62, except that the output from Machine 4 would not go back upstream, but would go to a separate buffer altogether. Such an arrangement would effectively remove a certain proportion of the material each interval and deposit it in this separate buffer.

If no other information is sought, this buffer can be left as is, with no exit. Then, at the end of a run, its contents would represent the total amount of material (destructively) inspected.

If we want to also model detecting bad units while testing, we would simply add an inspection machine (and buffer) behind this separate buffer. What passes through this machine represents good inspected material. The differences between these two new buffers represents bad inspected material.

Of course, several such inspection arrangements can be modeled and any number of them can be made to use the same off-line buffer.

Most often, though, sampling inspection is not destructive, and those units which pass the test are brought back on line, usually at the same point where the non-inspected units reside. Such an arrangement can easily be modeled in GENMOD. Figure 63 shows such a configuration.

Machine 1 does the work. Machines 2 and 3 are proportional splitters, with the latter "sampling" 5% (1 out of 20) of the material. Buffer 3 holds the units to be inspected. Machine 4 does the inspection, rejecting on the average of 2%. The non-defective units go back on-line at Buffer 4. The defective units disappear. At the end of the run, the total parts handled by Machine 4 represents the total pieces inspected. This arrangement, though, will <u>not</u> show total pieces rejected. If that information is sought, add another buffer behind Machine 4 and a perfect machine behind that. This latter machine will then pass the material back to the line and the difference between its total production and that of Machine 4 represents total parts rejected by the inspection process.

Remember, in this instance, that the inspector will on the average reject 2% of 5%, or only <u>.1% of all material</u>. Be sure to calculate proportions and ratios properly.

Note also that no matter how many steps are consumed by the inspection arm of this model, the original non-sampled units are immediately available in Buffer 4 for whatever operation follows. This is a correct usage of unequal parallel branches.

3.7.7 <u>Shuffling Machine Cards</u>

The discussion on renumbering machines stated that the user has the option of specifying the first machine number in his model, but that all machines from there on must be numbered consecutively. This is always true. However, it does not imply that the <u>data cards</u> must be presented consecutively.

The user has a further option available. As long as a machine card has a number punched on it and as long as the total number of cards matches the count of machines, then

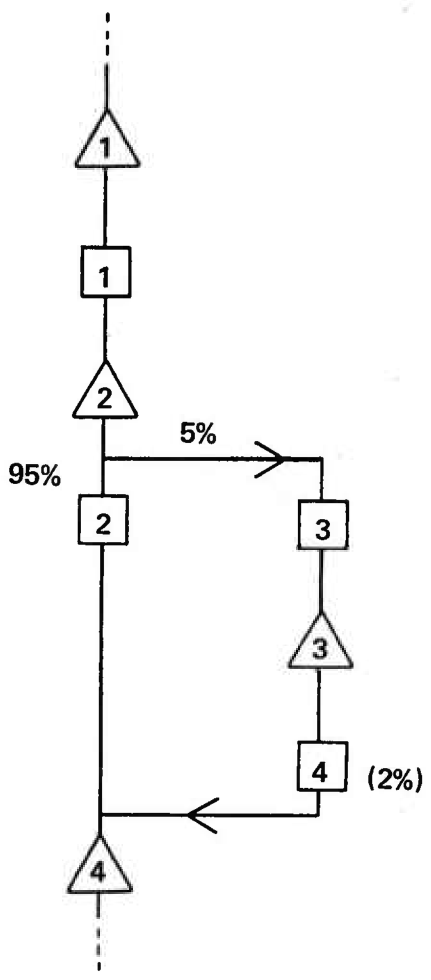


Figure 63 — Sampled Inspection With Return

the machine cards can be read in any order whatsoever (except when grouped components are involved).

This arrangement is useful for several reasons, the least of which is to make it easier to read a deck of cards that has fallen on the floor and gotten out of order. Many times it is convenient to document a model with a listing of the input, as we have done. Also, in complex models with many parallel branches, it is often easier to follow such a listing if the cards are arranged by branch rather than in strict numerical order. This option allows the cards to be sorted in any convenient manner.

Another benefit from this technique is the flexibility it lends to the punching process itself. Very quickly the reader will notice that most punched information for like machines is identical. Usually only the first five variables on the card change and the rest can be duplicated. By means of a drum control card, a GENMOD deck can be punched much faster than imagined, especially if all like machines are punched one after the other. The cards can then be left in the order in which they were punched.

The only restriction to ordering involves grouped components (Types 5 and 6). Each group must be presented as a unit, with the peripherals (+5, ±6) appearing immediately before the corresponding operational element (-5).

If it is too confusing, the reader should be aware of one more trick. It is not even necessary to punch any machine numbers on the machine cards. They can be left off, if the user wishes to save punching (or re-punching when sub-models are combined). When no machine number appears on a card, GENMOD automatically assigns that machine the next highest number. It is clear, then, that if machine numbers are left off, the cards must be read in consecutive order.

In summary, then, two main options are available for punching and reading machine cards.

a. Without numbers = The cards must then be put in consecutive order and will be numbered automatically, beginning with the value specified on the machine count card.

b. With numbers = The cards can be read in any order, except for grouped components.

Note that regardless of what manner is used for punching and reading, the first page of the GENMOD printout will always list the machines in numerical order (deleting Type 99's).

3.7.8 External Entry Points

Every GENMOD model must begin with a Buffer 0, regardless of how the other buffers are numbered. This point is always considered the main entry for raw material onto the line. The amount of such raw material available each and every interval is specified on the environment card.

However, Buffer 0 is not the only point by which material can be introduced onto a line. Technically, every buffer can be used to bring parts onto a line. This is done very easily, so easily in fact, that the user must be sure of what he is doing.

The initial discussion on buffer cards (page 48) showed four variables being read from each card. GENMOD actually reads one additional variable (located in columns 29-38). If this field is left blank, or 0.0 as in all examples shown thus far, that buffer is assumed to be a regular in-line buffer. If, however, any positive quantity is placed in that field, that buffer is assumed to be an external entry point and will act differently than a normal buffer.

Quite simply, at the start of each and every interval such entry points are loaded with the quantity specified, just as Buffer 0 is loaded with the amount of raw material specified. All previous contents are wiped out, i.e. material cannot be accumulated in an external entry point, just as material cannot be accumulated in Buffer 0. Thus, no machine should ever be directed to deposit material in an external entry point. The transfer will take place, but the material will immediately be wiped out.

Figure 64 shows a model using such external entry points and Figure 65 shows an input deck for that model.

The circled buffers (1,5,6,and 9) are external entries for sub-parts, which can be off the shelf, made by other sub-lines, etc. Note that no machines enter them and they are located at various levels on the model.

The data cards for these buffers clearly show the extra variable (20.0 in all cases). Since these buffers never receive parts from any machines, there is no real need for a maximum capacity, shown here to be the same as the external supply. Note that the Environment Card still contains a value (20.0) for the raw material available to Buffer 0.

Figure 66 shows the control point portion of the first page of the printout for a run of this model. Note the remarks identifying these four buffers as external entry points and the fact that these buffers have no entries.

No track is kept of how much material is actually brought in from external supplies. This can usually be determined by checking the total material handled by the machines which exit these points. The value printed as total raw material consumed on a run only concerns the material entering Buffer 0.

Because the parts entering through an external point are invariably sub-parts awaiting assembly, external entry points usually are associated with a Type 3 machine somewhere downstream, such as Machines 4, 7 and 9 above. Such an arrangement is not, of course, mandatory as long as the user knows what he is doing.

Lastly, since parts can be brought onto the mainline at any point and at any time, there may, or may not be, matching parts from the main line available. These sub-parts, then, might accumulate in a buffer for quite some time. In this example, the material brought in from Buffers 6 and 9 will accumulate in Buffers 10 and 12 until matching parts reach Buffer 11.

Remember that these external entry points will be fed by the specific amount of material every interval and these parts immediately become available to the exiting machines.

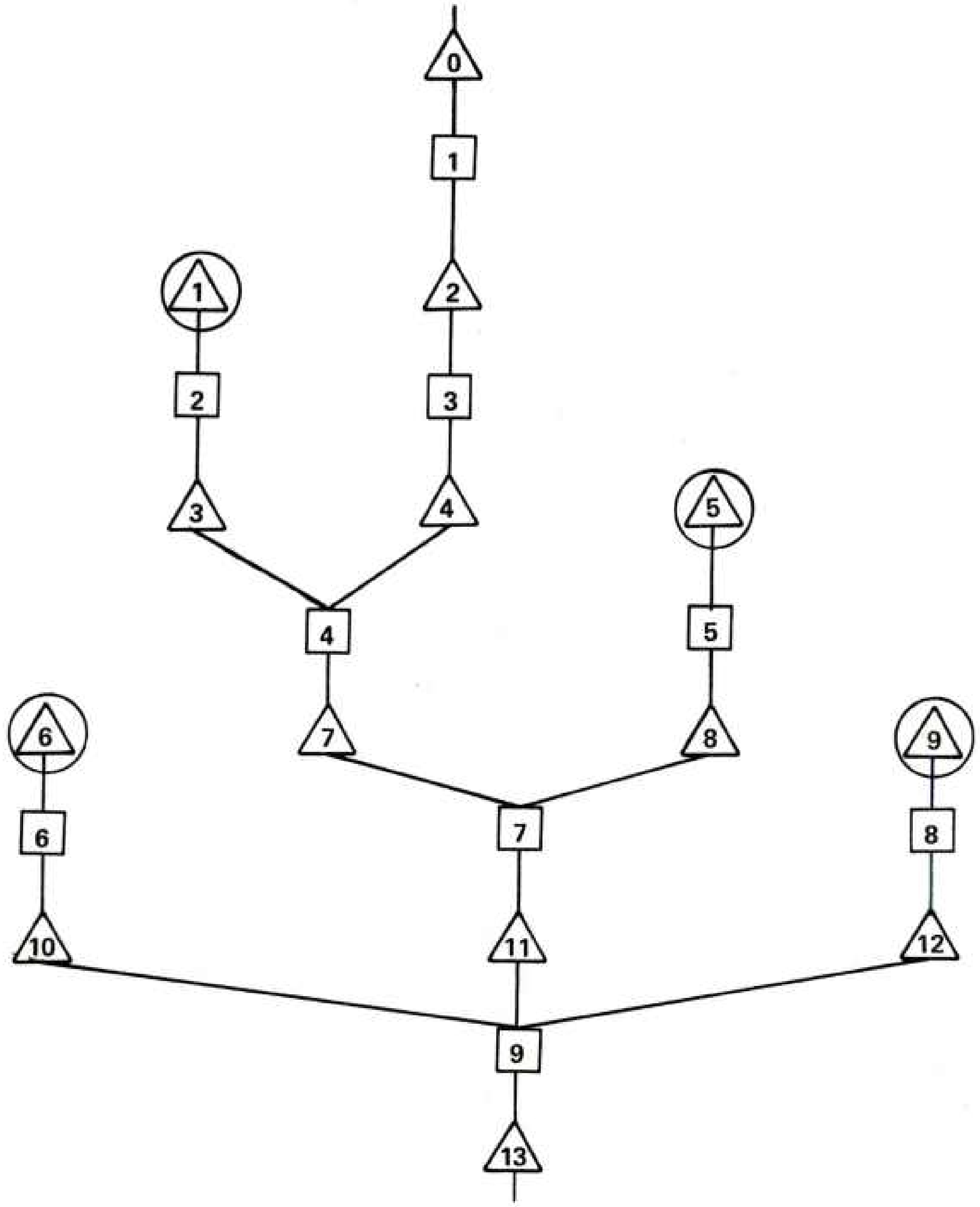


Figure 64 – Multiple Entry Model

```
MULTIPLE ENTRY TEST CASE
009
STEP1      STEP2A     STEP2B     STEP3A     STEP3B     STEP4A     STEP4B    STEP4C
STEP5
009
  1  1  1  0  2   20.00   40.00   8.000   4.000   0.000   0.000  1  0
  2  2  1  1  3   20.00   40.00   8.000   4.000   0.000   0.000  1  0
  3  3  1  2  4   20.00   40.00   8.000   4.000   0.000   0.000  1  0
  4  4  1  3  7   40.00   40.00   8.000   4.000   0.000   0.000 -3  2
  5  5  1  5  8   20.00   40.00   8.000   4.000   0.000   0.000  1  0
  6  6  1  6 10   20.00   40.00   8.000   4.000   0.000   0.000  1  0
  7  7  1  7 11   40.00   40.00   8.000   4.000   0.000   0.000 -3  2
  8  8  1  9 12   20.00   40.00   8.000   4.000   0.000   0.000  1  0
  9  9  1 10 13   60.00   40.00   8.000   4.000   0.000   0.000 -3  3
013
  1   0.00000   20.0000    1   20.0000
  2   0.00000  100.0000    1
  3   0.00000  100.0000    1
  4   0.00000  100.0000    1
  5   0.00000   20.0000    1   20.0000
  6   0.00000   20.0000    1   20.0000
  7   0.00000   100.000    1
  8   0.00000   100.000    1
  9   0.00000   20.0000    1   20.0000
 10   0.00000   100.000    1
 11   0.00000   100.000    1
 12   0.00000   100.000    1
 13   0.00000   0.00000    1
   20.0000   420.000    0    0    0    0   1.00000
0000000000000000000000
```

Figure 65 — Multiple Entry Input Deck

```
CONTROL POINT DATA
------------------

CONTROL  INITIAL   MAXIMUM   STATUS                     MAXIMUM  MAXIMUM
 POINT   CONTENTS  CAPACITY  PRINT   ENTRIES  EXITS     INPUT    OUTPUT
-------  --------  --------  ------  -------  -----    -------  -------
   1        0.         20.    YES       0       1         0.0     20.0   (ENTRY FOR 20.0 PARTS)
   2        0.        100.    YES       1       1        20.0     20.0
   3        0.        100.    YES       1       1        20.0     20.0
   4        0.        100.    YES       1       1        20.0     20.0
   5        0.         20.    YES       0       1         0.0     20.0   (ENTRY FOR 20.0 PARTS)
   6        0.         20.    YES       0       1         0.0     20.0   (ENTRY FOR 20.0 PARTS)
   7        0.        100.    YES       1       1        20.0     20.0
   8        0.        100.    YES       1       1        20.0     20.0
   9        0.         20.    YES       0       1         0.0     20.0   (ENTRY FOR 20.0 PARTS)
  10        0.        100.    YES       1       1        20.0     20.0
  11        0.        100.    YES       1       1        20.0     20.0
  12        0.        100.    YES       1       1        20.0     20.0
  13        0.    ********    YES       1       0        20.0      0.0   (OUTPUT POINT)
```

Figure 66 — Multiple Entry Control Point Data

3.7.9 Purge and Clear Option

Table 5 presented the values available for variables IPURGE and ICLEAR on the Environment Card, but they have not been discussed yet. Their main purpose is to dump all material from all buffers under specified conditions.

If the Purge option is invoked, all buffers will be reset to zero whenever any machine on the line fails. This feature was added to allow for the dumping of contaminated material caused by the failure of a chemical line. It could, of course, be used for any such purpose. An account is maintained of how much material winds up being dumped through use of this option.

If the Clear option is invoked, all buffers will be reset to zero after passage of a fixed number of intervals (IDUR). This feature can be used to dump material at the end of a shift or of a week's production, etc. It, too, is designed mostly for chemical lines, which cannot leave in-process material lying around overnight or over a weekend. The value of IDUR is in intervals and should be calculated to correspond to the proper duration of real time. No account is kept of how much material is cleared by this option. Such information, though, can be gotten by printing the buffer status table at a matching frequency and summing the buffers.

Neither the Purge nor Clear options apply to the output buffer.

The version of GENMOD discussed here will carry out a complete purge or clear, i.e. all buffers are dumped. Other versions of GENMOD allow for setting up a prioritized system of purging and clearing, whereby only selected buffers are effected under specific conditions. Setting up such a model requires careful planning. Information concerning this version is available upon request.

3.7.10 Maintenance Policy (Operation Model)

All examples shown so far were for IMODE=0, i.e. continuous simulation. This meant that the simulation was run without stopping. It does not mean, however, that the process being simulated ran continuously. As long as all buffers on a line are untouched during a break and no machines are adjusted or repaired, then when the line is restarted everything picks up from where it left off, as if there were no stoppage at all. Thus, a Mode 0 run of 1000 intervals merely represents 1000 consecutive real time intervals, which could have been broken up in any manner whatsoever.

But, if during a scheduled break or after a shift, it is planned that buffers are cleared or machines adjusted, then a simple Mode 0 run is not appropriate. The question of clearing buffers was discussed in the last section. Adjusting machines will be discussed now.

Just as IDUR can be used to signal a clearing of buffers (if ICLEAR≠0), so too it can signal an adjustment to machines (if IMODE≠0). If maintenance men are to be employed during stoppages, their prime concern would probably be for repairing machines which are down, i.e. performing corrective maintenance. Thus, if IMODE=1, every IDUR intervals GENMOD will pause and all downed machines will be brought back up and a new time-to-next failure sampled for those machines only. This version of GENMOD does not discriminate against machines with long times-to-repair or make allowances for the number of repairmen actually available. It assumes that sufficient men are available and that the break is sufficiently long to repair all machines, regardless of the scheduled time-to-repair.

IMODE=2 will carry this maintenance action one step further. In addition to repairing all machines, GENMOD will also calculate a new time-to-next failure for all machines. This provision satisfies those who feel that preventive maintenance, in which machines are actually adjusted, changes their character to the extent that their time-to-next failure has been altered. This option is really no more than an exercise, since the failure times currently stored for all machines are already a random function and nothing is gained by calculating them again. In fact, there is a 50/50 chance that the new failure time will be shorter than the current one, since GENMOD makes no attempt to "optimize" failure times.

3.8 Sequenced Operations

An abbreviated example will now be presented of a model using sequenced operations, Type -2. Figure 67 shows this model, with the manner I use to diagram Type 2 components, namely a large block segmented into several separate machines.

A typical input deck for this model is given as Figure 68. Note first that I have deliberately defined five category titles, but have only used four of them. The effect of this extra, unused category will be seen shortly. Machines 3 through 8 are the alternating operations and the AUX field is used to indicate the cycle duration, in this case three. These six machines are distinguished as two different groups by their differing sets of input and output buffers.

A portion of the first input page for a run of this model is shown as Figure 69. Of chief importance on this page is the indication that Type 2 elements are present in the model, namely the section entitled "Tandem Components". Here, the first and last element of each set of such machines are listed along with the corresponding output buffer. Note carefully that in order for a block of Type 2 machines to work properly, the machines within the block must be numbered consecutively.

It is not intended for a thorough examination of this run to be made. Therefore, only the major sections are being discussed, to alert the user to features pertaining to Type 2's not seen before.

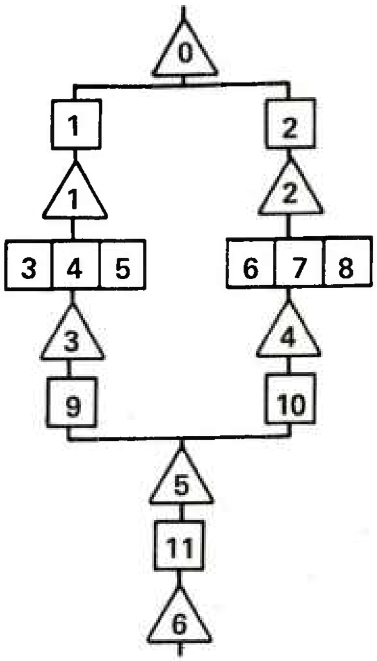


Figure 67 — Sequenced Operations Model

For example, the periodic machine handling table clearly shows that Machines 3, 4, and 5 do indeed take turns working, each handling 20 parts every third interval. Machines 6, 7, and 8 work likewise. The periods of eleven and six parts during Intervals 23 and 24 correspond to a failure of Machine 2, rather than to failures of the alternating machines themselves.

On this particular run, two of the alternators did indeed fail. But, neither failure occurred when the machine was scheduled to begin its cycle and both failures were so short that the machines were repaired before they were to begin the next cycle. Note that the values shown on this table are the parts handled during the first interval of each alternator's three-step cycle. If an alternator happens to go down during this first interval, its production will generally be less than its normal share. If the failure, though, occurs within the cycle, the material will immediately be transferred to the output buffer, if there is room. In either case, the machine will not start another cycle until the first time it is routinely called upon, following its restoration to service.

The summary table for availability by category is shown, as Figure 71, for the sole purpose of presenting the note that accompanies a category title that is not used in a model.

```
0000000000000000000000
ALTERNATING OPERATIONS
005
OPER1       ALTERNATOROPER3      OPER4      OPER5
011
  1   1   1   0   1    20.00    30.00    6.000    4.000    0.000    0.000   0   0
  2   1   2   0   2    20.00    30.00    6.000    4.000    0.000    0.000   0   0
  3   2   1   1   3    22.00    30.00    6.000    4.000    0.000    0.000  -2   3
  4   2   2   1   3    22.00    30.00    6.000    4.000    0.000    0.000  -2   3
  5   2   3   1   3    22.00    30.00    6.000    4.000    0.000    0.000  -2   3
  6   2   4   2   4    22.00    30.00    6.000    4.000    0.000    0.000  -2   3
  7   2   5   2   4    22.00    30.00    6.000    4.000    0.000    0.000  -2   3
  8   2   6   2   4    22.00    30.00    6.000    4.000    0.000    0.000  -2   3
  9   3   1   3   5    25.00    30.00    6.000    4.000    0.000    0.000   1   0
 10   3   2   4   5    25.00    30.00    6.000    4.000    0.000    0.000   1   0
 11   4   1   5   6    55.00    30.00    6.000    4.000    0.000    0.000   1   0
006
  1    0.00000    100.000    1
  2    0.00000    100.000    1
  3    0.00000    100.000    1
  4    0.00000    100.000    1
  5    0.00000    100.000    1
  6    0.00000    0.00000    1
   40.0000    420.000    0    0    0    0    1.00000
0000000000000000000000
   30     1     1     2     0
0000000000000000000000
```

Figure 68 – Sequenced Operations Input Deck

No harm is done by defining more titles than are used. This often saves repunching an input deck following certain modifications. The seeming error that appears for OPER3, i.e. one failure, yet an average availability of 1.0 is due to the fact that the machine failed during the last interval of simulation. Total up and down times are calculated based upon the status of each machine at the <u>beginning</u> of each interval. Thus, this machine was <u>up</u> at the start of the final interval and the run was terminated before any downtime was counted. Yet, the failure is still recorded.

Only a portion of the periodic buffer status table is given in Figure 72. There are two main things to note here. Although the first of the alternators, Machines 4 and 7, processed their parts in the second interval, nothing appears in their respective output buffers until the

```
        ALTERNATING OPERATIONS

                           INPUT  OUTPUT           FAILURE       REPAIR     NOMINAL  COMP   AUX   DEFECT
MACHINE  OPERATION(SEQ)   BUFFER  BUFFER   RATE   MEAN  SIGMA   MEAN  SIGMA  AVAIL.  TYPE  FIELD   RATE
-------  --------------   ------  ------   ----   ----  -----   ----  ------ ------  ----  -----  ------
   1     OPER1     ( 1)      0       1     20.0   30.0    6.0    4.0    0.0  .8824     0     0   0.00000
   2     OPER1     ( 2)      0       2     20.0   30.0    6.0    4.0    0.0  .8824     0     0   0.00000
   3     ALTERNATOR( 1)      1       3     22.0   30.0    6.0    4.0    0.0  .8824    -2     3   0.00000
   4     ALTERNATOR( 2)      1       3     22.0   30.0    6.0    4.0    0.0  .8824    -2     3   0.00000
   5     ALTERNATOR( 3)      1       3     22.0   30.0    6.0    4.0    0.0  .8824    -2     3   0.00000
   6     ALTERNATOR( 4)      2       4     22.0   30.0    6.0    4.0    0.0  .8824    -2     3   0.00000
   7     ALTERNATOR( 5)      2       4     22.0   30.0    6.0    4.0    0.0  .8824    -2     3   0.00000
   8     ALTERNATOR( 6)      2       4     22.0   30.0    6.0    4.0    0.0  .8824    -2     3   0.00000
   9     OPER3     ( 1)      3       5     25.0   30.0    6.0    4.0    0.0  .8824     1     0   0.00000
  10     OPER3     ( 2)      4       5     25.0   30.0    6.0    4.0    0.0  .8824     1     0   0.00000
  11     OPER4     ( 1)      5       6     55.0   30.0    6.0    4.0    0.0  .8824     1     0   0.00000

TANDEM COMPONENTS
-----------------

          SET   FIRST ELEMENT   LAST ELEMENT   OUTPUT BUFFER
          ---   -------------   ------------   -------------
           1          3               5               3
           2          6               8               4
```

Figure 69 — Example 3.7 — Page 1

```
ALTERNATING OPERATIONS

PERIODIC MACHINE HANDLING
-------------------------
            1      2      3      4      5      6      7      8      9     10     11
   1       20.    20.     0.     0.     0.     0.     0.     0.     0.     0.     0.
   2       20.    20.     0.    20.     0.     0.    20.     0.     0.     0.     0.
   3       20.    20.     0.     0.    20.     0.     0.    20.     0.     0.     0.
   4       20.    20.    20.     0.     0.    20.     0.     0.     0.     0.     0.
   5       20.    20.     0.    20.     0.     0.    20.     0.    20.    20.     0.
   6       20.    20.     0.     0.    20.     0.     0.    20.    20.    20.    40.
   7       20.    20.    20.     0.     0.    20.     0.     0.    20.    20.    40.
   8       20.    20.     0.    20.     0.     0.    20.     0.    20.    20.    40.
   9       20.    20.     0.     0.    20.     0.     0.    20.    20.    20.    40.
  10       20.    20.    20.     0.     0.    20.     0.     0.    20.    20.    40.
  11       20.    20.     0.    20.     0.     0.    20.     0.    20.    20.    40.
  12       20.    20.     0.     0.    20.     0.     0.    20.    20.    20.    40.
  13       20.    20.    20.     0.     0.    20.     0.     0.    20.    20.    40.
  14       20.    20.     0.    20.     0.     0.    20.     0.    20.    20.    40.
  15       20.    20.     0.     0.    20.     0.     0.    20.    20.    20.    40.
  16       20.    20.    20.     0.     0.    20.     0.     0.    20.    20.    40.
  17       20.    20.     0.    20.     0.     0.    20.     0.    20.    20.    40.
  18       20.    20.     0.     0.    20.     0.     0.    20.    20.    20.    40.
  19       20.    20.    20.     0.     0.    20.     0.     0.    20.    20.    40.
  20       20.    20.     0.    20.     0.     0.    20.     0.    20.    20.    40.
  21       20.    20.     0.     0.    20.     0.     0.    20.    20.    20.    40.
  22       20.    11.    20.     0.     0.    20.     0.     0.    20.    20.    40.
  23       20.     6.     0.    20.     0.     0.    11.     0.    25.    20.    40.
  24       20.    20.     0.     0.    20.     0.     0.     6.    25.    20.    45.
  25       20.    20.    20.     0.     0.    20.     0.     0.    10.    20.    45.
  26       20.    20.     0.    20.     0.     0.    20.     0.    20.    11.    30.
  27       20.    20.     0.     0.    20.     0.     0.    20.    20.     6.    31.
  28       20.    20.    20.     0.     0.    20.     0.     0.    20.    20.    26.
  29       20.    20.     0.    20.     0.     0.    20.     0.    20.    25.    40.
  30       20.    20.     0.     0.    20.     0.     0.    20.     0.    25.    45.
```

Figure 70 — Example 3.7 — Page 2

```
AVAILABILITY DATA           (BY CATEGORY)
-----------------
                     NO. OF    TOTAL     TOTAL      TOTAL      AVERAGE      AVERAGE
        CATEGORY    MACHINES  FAILURES  DOWN TIME  UP TIME  AVAILABILITY  STOP TIME
        --------    --------  --------  ---------  -------  ------------  ---------
( 1) OPER1              2         1         1         59        .9833         0.0
( 2) ALTERNATOR         6         2         2        178        .9889        11.0
( 3) OPER3              2         1         0         60       1.0000         4.0
( 4) OPER4              1         0         0         30       1.0000         5.0
( 5) OPER5              0    (NOT PRESENT IN DESIGN)

        4 TOTAL FAILURES REQUIRED        3 INTERVALS REPAIR TIME
     AVERAGE REPAIR TIME=      .75

     SYSTEM WAS DEFECT-FREE FOR    27 INTERVALS OUT OF     30
     DEMAND FOR MAINTENANCE ACTION= .1000
```

Figure 71 — Example 3.7 — Page 3

```
                ALTERNATING OPERATIONS

                PERIODIC BUFFER STATUS
                ----------------------

INTERVAL          1       2       3       4       5       6
---------------------------------------------------------------
    1            20.     20.      0.      0.      0.      0.
---------------------------------------------------------------
    2            20.     20.      0.      0.      0.      0.
---------------------------------------------------------------
    3            20.     20.      0.      0.      0.      0.
---------------------------------------------------------------
    4            20.     20.     20.     20.      0.      0.
---------------------------------------------------------------
    5            20.     20.     20.     20.     40.      0.
---------------------------------------------------------------
    6            20.     20.     20.     20.     40.     40.
---------------------------------------------------------------
    7            20.     20.     20.     20.     40.     80.
---------------------------------------------------------------
    8            20.     20.     20.     20.     40.    120.
---------------------------------------------------------------
    9            20.     20.     20.     20.     40.    160.
---------------------------------------------------------------
   10            20.     20.     20.     20.     40.    200.
---------------------------------------------------------------
   11            20.     20.     20.     20.     40.    240.
---------------------------------------------------------------
   12            20.     20.     20.     20.     40.    280.
---------------------------------------------------------------
   13            20.     20.     20.     20.     40.    320.
---------------------------------------------------------------
   14            20.     20.     20.     20.     40.    360.
---------------------------------------------------------------
   15            20.     20.     20.     20.     40.    400.
---------------------------------------------------------------
   16            20.     20.     20.     20.     40.    440.
---------------------------------------------------------------
   17            20.     20.     20.     20.     40.    480.
---------------------------------------------------------------
   18            20.     20.     20.     20.     40.    520.
---------------------------------------------------------------
   19            20.     20.     20.     20.     40.    560.
---------------------------------------------------------------
   20            20.     20.     20.     20.     40.    600.
---------------------------------------------------------------
   21            20.     20.     20.     20.     40.    640.
---------------------------------------------------------------
   22            20.     11.     20.     20.     40.    680.
---------------------------------------------------------------
   23            20.      6.     35.     20.     45.    720.
---------------------------------------------------------------
   24            20.     20.     10.     20.     45.    765.
---------------------------------------------------------------
   25            20.     20.     20.     11.     30.    810.
---------------------------------------------------------------
   26            20.     20.     20.      6.     31.    840.
---------------------------------------------------------------

INTERVAL          1       2       3       4       5       6
```

Figure 72 — Example 3.7 — Page 4

fourth interval. The steps of a three-interval cycle of operation can be thought of as load/process/unload. Thus, the parts are normally released in the interval preceding the next cycle, which in this case would begin in Interval 5. Note also the sudden jump in the contents of Buffer 3 at Interval 23. This is a result of the release of the in-process parts upon the failure of Machine 3 during that interval.

3.9 Assembly, Splits, and Shifted Numbering

This next illustration will take care of several loose ends at one time. It involves proportional splitters, assembly machines, and crossed lines, all with non-standard numbering. Figure 73 shows the diagram for this line.

Note several interesting features of this model:

a. The machines are numbered starting at eight, rather than one.

b. The first three machines are designed to be perfect distributors of the raw material, sending 1/3 consistently to each of the three branches.

c. The assembly machines (17 through 20) are diagramed as receiving parts from more than one input point which indeed they do. Assembly machines are the only GENMOD blocks that draw from more than one input node.

d. The middle buffers (4 through 9) are deliberately numbered out of sequence, not arbitrarily, but because the input buffers to the subsequent assembly machines must be numbered sequentially.

e. The lines of flow for the first time cross each other, as they will on the actual constructed line. There is nothing sacred about maintaining purely straight lines in a model.

The incorporation of all these features into the punched deck is shown clearly in Figure 74. Note especially that:

a. The machine count card ("013 8") contains the starting number of the first machine, as explained in para. 3.7.3.

b. Machines 8-10 are first designated as perfect machines by having a zero MTTR. They are further specified as proportional splitters by having 0.33333 in the DEF field. Always remember that a perfect machine with a "defect" rate is automatically considered a proportional splitter.

c. The AUX field for the assemblers contains the number of buffers each draws from and the particular buffer specified as IN is in all cases the first of such buffers. Also, the speed of these machines is rated in terms of total input parts, rather than output parts.

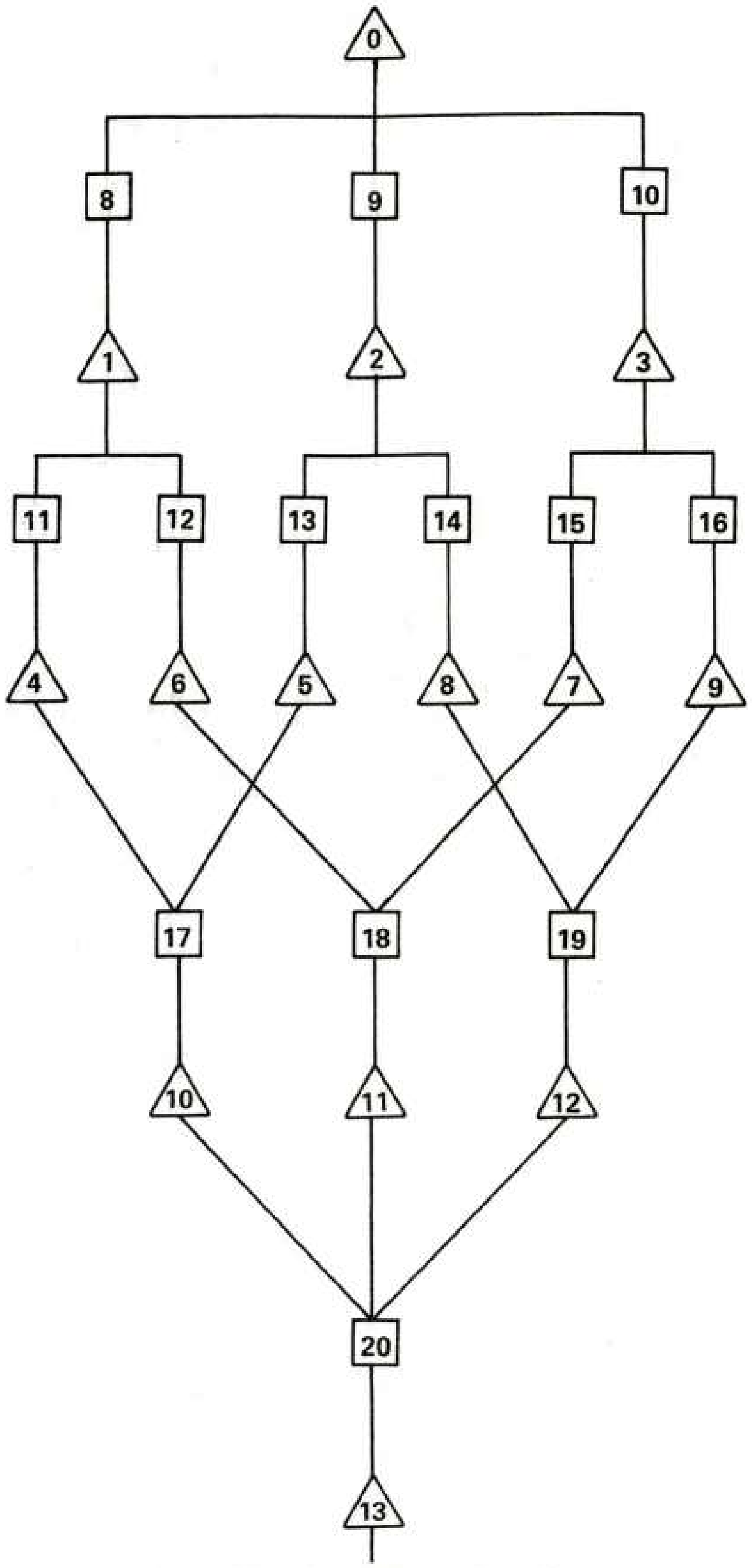


Figure 73 – Assembly with Crossing

```
0000000000000000000000
ASSEMBLY WITH STAGGERED NUMBERING
004
SPLIT      STEP1      ASSEMBLE1 ASSEMBLE2
013  8
  8  1  1  0  1   20.00   0.000   0.000   0.000   0.000 0.33333  0  0
  9  1  2  0  2   20.00   0.000   0.000   0.000   0.000 0.33333  0  0
 10  1  3  0  3   20.00   0.000   0.000   0.000   0.000 0.33333  0  0
 11  2  1  1  4   10.00   30.00   6.000   3.000   0.000   0.000  0  0
 12  2  2  1  6   10.00   30.00   6.000   3.000   0.000   0.000  0  0
 13  2  3  2  5   10.00   30.00   6.000   3.000   0.000   0.000  0  0
 14  2  4  2  8   10.00   30.00   6.000   3.000   0.000   0.000  0  0
 15  2  5  3  7   10.00   30.00   6.000   3.000   0.000   0.000  0  0
 16  2  6  3  9   10.00   30.00   6.000   3.000   0.000   0.000  0  0
 17  3  1  4 10   24.00   30.00   6.000   3.000   0.000   0.000 -3  2
 18  3  2  6 11   24.00   30.00   6.000   3.000   0.000   0.000 -3  2
 19  3  3  8 12   24.00   30.00   6.000   3.000   0.000   0.000 -3  2
 20  4  1 10 13   39.00   30.00   6.000   3.000   0.000   0.000 -3  3
013
  1     0.00000  100.000      1
  2     0.00000  100.000      1
  3     0.00000  100.000      1
  4     0.00000  100.000      1
  5     0.00000  100.000      1
  6     0.00000  100.000      1
  7     0.00000  100.000      1
  8     0.00000  100.000      1
  9     0.00000  100.000      1
 10     0.00000  100.000      1
 11     0.00000  100.000      1
 12     0.00000  100.000      1
 13     0.00000  0.00000      1
   60.0000    420.000    0     0     0     0    1.00000
0000000000000000000000
   30     1     1     2    0
0000000000000000000000
```

Figure 74 — Assembly Line Input Deck

A portion of the first output page is shown in Figure 75. The machines are indeed identified as 8 through 20, although internally they are numbered 1 through 13. Whenever machine numbers are referred to by a GENMOD printout they will always be in terms of the user's numbering scheme. The places where such references appear include the input listing, times to first failure, periodic machine handling, total parts handled, availability summary, and logbook.

The nominal availability for the dummy machines (8-10) is shown as 1.000, reflecting their logical perfection. This is further evidenced by the times to first failure for these same machines. This value, theoretically infinite, is shown simply as ******.

The figure only contains the first line of the times to first failure table and only the first seven machines are shown. This is done only because of the size limitation of this manual. In actuality, the printout lists eight machines to a line, but such a line is too long to reproduce here. Similarly, the periodic machine handling table lists 17 machines per line and the periodic buffer status table can hold 15 buffers per line. Both of these lines are too large and is the prime reason why only an excerpted buffer status table is shown for this example.

In examining the periodic machine handling table for this model we are chiefly interested in the number of parts handled by the assemblers. As stated in the original description of Type 3 components, their speed must be specified in terms of total pieces handled. Also, no more material will be handled than is contained in any single input buffer. This is shown quite clearly in the table during the intervals related to the failure of Machine 14.

This machine went down in Interval 10. Prior to that time, Machines 11 through 16 were all handling ten parts per interval and Buffers 4 through 9 were thus all being fed ten parts per interval. This allowed the first bank of assemblers, 17-19, to handle $\underline{20}$ parts each. These 20 sub-parts, though, are deposited in Buffers 10-12 as $\underline{10}$ modules in each. These $\underline{30}$ modules are then handled by the final assembly (20) and deposited in the output as $\underline{10}$ finished parts.

During the 10th interval, though, only about 3.35 parts were processed by Machine 14. (This value was derived by consulting the logbook.) Thus, in the eleventh interval, Machine 19 could only handle that much from each of its input buffers, accounting for the value of 7. (actually 6.7) shown in the table. These parts were passed to Buffer 12 as 3.35 modules, which forced the final machine to handle that much from each of its three buffers in the twelfth interval.

Following that, Machine 14 stopped producing until the 19th interval. During that time, Machine 19 had to stop producing because there was never anything in Buffer 8 for it to match with Buffer 9. The other two assemblers on this level, though, continued to run

```
          ASSEMBLY WITH STAGGERED NUMBERING
                                  INPUT   OUTPUT            FAILURE          REPAIR      NOMINAL   COMP   AUX    DEFECT
MACHINE   OPERATION(SEQ)          BUFFER  BUFFER    RATE    MEAN   SIGMA    MEAN  SIGMA   AVAIL.    TYPE  FIELD   RATE
-------   ---------------         ------  ------    ----    ----   -----    ----  -----  -------    ----  -----  ------
    8     SPLIT      ( 1)           0       1       20.0     0.0    0.0      0.0   0.0   1.0000      0      0    .33333
    9     SPLIT      ( 2)           0       2       20.0     0.0    0.0      0.0   0.0   1.0000      0      0    .33333
   10     SPLIT      ( 3)           0       3       20.0     0.0    0.0      0.0   0.0   1.0000      0      0    .33333
   11     STEP1      ( 1)           1       4       10.0    30.0    6.0      3.0   0.0    .9091      0      0   0.00000
   12     STEP1      ( 2)           1       6       10.0    30.0    6.0      3.0   0.0    .9091      0      0   0.00000
   13     STEP1      ( 3)           2       5       10.0    30.0    6.0      3.0   0.0    .9091      0      0   0.00000
   14     STEP1      ( 4)           2       8       10.0    30.0    6.0      3.0   0.0    .9091      0      0   0.00000
   15     STEP1      ( 5)           3       7       10.0    30.0    6.0      3.0   0.0    .9091      0      0   0.00000
   16     STEP1      ( 6)           3       9       10.0    30.0    6.0      3.0   0.0    .9091      0      0   0.00000
   17     ASSEMBLE1  ( 1)           4      10       24.0    30.0    6.0      3.0   0.0    .9091     -3      2   0.00000
   18     ASSEMBEL1  ( 2)           6      11       24.0    30.0    6.0      3.0   0.0    .9091     -3      2   0.00000
   19     ASSEMBLE1  ( 3)           8      12       24.0    30.0    6.0      3.0   0.0    .9091     -3      2   0.00000
   20     ASSEMBLE2  ( 1)          10      13       39.0    30.0    6.0      3.0   0.0    .9091     -3      3   0.00000

TIMES OF FIRST FAILURE
----------------------
 (  8=******)   (  9=******)   ( 10=******)   ( 11=     33)   ( 12=     33)   ( 13=     27)   ( 14=     10)
-----------------------------------------------------------------------------------------------------------------
```

Figure 75 — Example 3.9 — Page 1

since their supplies of sub-parts were uninterrupted. The final machine, however, could not function because Buffer 12 was always empty, even though Buffers 10 and 11 kept filling.

Given that Machine 14 handled 4.6 parts when it came up during Interval 19, the reader should be able to trace the other associated values.

The variations in speed exhibited by Machines 9 and 16 were caused by their output buffers becoming full.

The last figure for this example shows an excerpted portion of the periodic buffer status table, with the first six buffers left off.

3.10 Control Devices

As our last test case, we shall return to the model shown in Figure 14, which showed an arrangement of Type 6 components (control devices) interspersed with Type 5 blocks.

An extract of a sample input deck was presented at that time. We shall now discuss a complete deck (Figure 78), which contains all variables for all machines as well as the category and buffer information.

Of prime importance when using Type 6 components is the fact that a card must be punched for each occurrence of each such machine. Thus, this model contains only 18 machines, but 22 machine cards, because all four control devices appear twice. Also, three points must be borne in mind for each of the Type 6 cards:

a. Each card should be numbered correctly and placed within the domain of its associated Type -5 machine. For example, Machines 8 and 9 are grouped with Machine 12.

b. The first card encountered for each control device must be punched as Type 6, with all subsequent cards punched as Type -6. Also, the first card must contain the correct failure and repair data. All subsequent cards need not.

c. The AUX of each card must contain the machine number of the associated Type -5 machine. Thus, the two cards for Machine 7 are punched "6 11" and "-6 13" in their TYPE and AUX fields.

Of course, all Type 6 machines should be shown as going to a dummy buffer, such as the 999 used here.

One other new feature has been included in this example. The BTI has been specified as five minutes, on the Environment Card. Thus, all speeds, failures, and repairs are assumed to be in terms of five minute intervals. The shift length is given as 84. (420 minutes). The

```
ASSEMBLY WITH STAGGERED NUMBERING

PERIODIC MACHINE HANDLING
--------------------------
              8      9     10     11     12     13     14     15     16     17     18     19     20
   1        20.    20.    20.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.
   2        20.    20.    20.    10.    10.    10.    10.    10.    10.     0.     0.     0.     0.
   3        20.    20.    20.    10.    10.    10.    10.    10.    10.    20.    20.    20.     0.
   4        20.    20.    20.    10.    10.    10.    10.    10.    10.    20.    20.    20.    30.
   5        20.    20.    20.    10.    10.    10.    10.    10.    10.    20.    20.    20.    30.
   6        20.    20.    20.    10.    10.    10.    10.    10.    10.    20.    20.    20.    30.
   7        20.    20.    20.    10.    10.    10.    10.    10.    10.    20.    20.    20.    30.
   8        20.    20.    20.    10.    10.    10.    10.    10.    10.    20.    20.    20.    30.
   9        20.    20.    20.    10.    10.    10.    10.    10.    10.    20.    20.    20.    30.
  10        20.    20.    20.    10.    10.    10.     3.    10.    10.    20.    20.    20.    30.
  11        20.    20.    20.    10.    10.    10.     0.    10.    10.    20.    20.     7.    30.
  12        20.    20.    20.    10.    10.    10.     0.    10.    10.    20.    20.     0.    10.
  13        20.    20.    20.    10.    10.    10.     0.    10.    10.    20.    20.     0.     0.
  14        20.    20.    20.    10.    10.    10.     0.    10.    10.    20.    20.     0.     0.
  15        20.    20.    20.    10.    10.    10.     0.    10.    10.    20.    20.     0.     0.
  16        20.    20.    20.    10.    10.    10.     0.    10.    10.    20.    20.     0.     0.
  17        20.    20.    20.    10.    10.    10.     0.    10.    10.    20.    20.     0.     0.
  18        20.    13.    20.    10.    10.    10.     0.    10.    10.    20.    20.     0.     0.
  19        20.    15.    20.    10.    10.    10.     5.    10.    10.    20.    20.     0.     0.
  20        20.    20.    20.    10.    10.    10.    10.    10.     8.    20.    20.     9.     0.
  21        20.    20.    20.    10.    10.    10.    10.    10.    10.    16.    16.    20.    14.
  22        20.    20.    20.    10.    10.    10.    10.    10.    10.    20.    20.    20.    30.
  23        20.    20.    20.    10.    10.    10.    10.    10.    10.    20.    20.    20.    30.
  24        20.    20.    20.    10.    10.    10.    10.    10.    10.    20.    20.    20.    30.
  25        20.    20.    20.    10.    10.    10.    10.    10.    10.    20.    20.    20.    30.
  26        20.    20.    20.    10.    10.    10.    10.    10.    10.    20.    20.    20.    30.
  27        20.    17.    20.    10.    10.     7.    10.    10.    10.    20.    20.    20.    30.
  28        20.    10.    20.    10.    10.     0.    10.    10.    10.    18.    20.    20.    30.
  29        20.    10.    20.    10.    10.     0.    10.     4.    10.     0.    20.    20.    30.
  30        20.    10.    20.    10.    10.     0.    10.     6.    10.     0.    12.    20.    27.
```

Figure 76 — Example 3.9 — Page 2

INTERVAL	7	8	9	10	11	12	13
1	0.	0.	0.	0.	0.	0.	0.
2	10.	10.	10.	0.	0.	0.	0.
3	10.	10.	10.	10.	10.	10.	0.
4	10.	10.	10.	10.	10.	10.	10.
5	10.	10.	10.	10.	10.	10.	20.
6	10.	10.	10.	10.	10.	10.	30.
7	10.	10.	10.	10.	10.	10.	40.
8	10.	10.	10.	10.	10.	10.	50.
9	10.	10.	10.	10.	10.	10.	60.
10	10.	3.	10.	10.	10.	10.	70.
11	10.	0.	17.	10.	10.	3.	80.
12	10.	0.	27.	17.	17.	0.	83.
13	10.	0.	37.	27.	27.	0.	83.
14	10.	0.	47.	37.	37.	0.	83.
15	10.	0.	57.	47.	47.	0.	83.
16	10.	0.	67.	57.	57.	0.	83.
17	10.	0.	77.	67.	67.	0.	83.
18	10.	0.	87.	77.	77.	0.	83.
19	10.	5.	97.	87.	87.	0.	83.
20	10.	10.	100.	97.	97.	5.	83.
21	12.	10.	100.	100.	100.	10.	88.
22	12.	10.	100.	100.	100.	10.	98.
23	12.	10.	100.	100.	100.	10.	108.
24	12.	10.	100.	100.	100.	10.	118.
25	12.	10.	100.	100.	100.	10.	128.
26	12.	10.	100.	100.	100.	10.	138.
INTERVAL	7	8	9	10	11	12	13

Figure 77 — Example 3.9 — Page 4

```
00000000000000000000000
TYPE 6 TEST CASE
008
STEP1      STEP2      CONTROL1  CONTROL2  CONTROL3  CONTROL4  STEP3      STEP4
018
  1  1  1  0  1   200.0   45.00   9.000   5.000   0.000   0.000  0  0
  2  1  2  0  2   200.0   45.00   9.000   5.000   0.000   0.000  0  0
  3  2  1  1  3   100.0   50.00   10.00   6.000   0.000   0.000  0  0
  4  2  2  1  4   100.0   50.00   10.00   6.000   0.000   0.000  0  0
  5  2  3  2  5   100.0   50.00   10.00   6.000   0.000   0.000  0  0
  6  2  4  2  6   100.0   50.00   10.00   6.000   0.000   0.000  0  0
  7  3  1  3999   0.000   60.00   12.00   5.000   0.000   0.000  6 11
  9  5  1  3999   0.000   60.00   12.00   5.000   0.000   0.000  6 11
 11  7  1  3  7   100.0   75.00   15.00   10.00   0.000 0.01500 -5  0
  8  4  1  4999   0.000   60.00   12.00   5.000   0.000   0.000  6 12
  9  5  1  4999   0.000   0.000   0.000   0.000   0.000   0.000 -6 12
 12  7  2  4  8   100.0   75.00   15.00   10.00   0.000 0.01500 -5  0
  7  3  1  5999   0.000   0.000   0.000   0.000   0.000   0.000 -6 13
 10  6  1  5999   0.000   60.00   12.00   5.000   0.000   0.000  6 13
 13  7  3  5  9   100.0   75.00   15.00   10.00   0.000 0.01500 -5  0
  8  4  1  6999   0.000   0.000   0.000   0.000   0.000   0.000 -6 14
 10  6  1  6999   0.000   0.000   0.000   0.000   0.000   0.000 -6 14
 14  7  4  6 10   100.0   75.00   15.00   10.00   0.000 0.01500 -5  0
 15  8  1  7 11   100.0   60.00   12.00   7.000   0.000   0.000  1  0
 16  8  2  8 11   100.0   60.00   12.00   7.000   0.000   0.000  1  0
 17  8  3  9 11   100.0   60.00   12.00   7.000   0.000   0.000  1  0
 18  8  4 10 11   100.0   60.00   12.00   7.000   0.000   0.000  1  0
011
  1    0.00000   1000.00    1
  2    0.00000   1000.00    1
  3    0.00000   1000.00    1
  4    0.00000   1000.00    1
  5    0.00000   1000.00    1
  6    0.00000   1000.00    1
  7    0.00000   1000.00    1
  8    0.00000   1000.00    1
  9    0.00000   1000.00    1
 10    0.00000   1000.00    1
 11    0.00000   0.00000    1
   400.000   84.0000     0     0     0     0   5.00000
00000000000000000000000
  120   12    0    0    0
00000000000000000000000
```

Figure 78 — Type 6 Input Deck

```
         TYPE 6 TEST CASE

                                   INPUT   OUTPUT               FAILURE          REPAIR       NOMINAL   COMP    AUX     DEFECT
MACHINE   OPERATION(SEQ)          BUFFER   BUFFER     RATE    MEAN   SIGMA     MEAN   SIGMA    AVAIL.    TYPE   FIELD     RATE
--------  --------------          ------   ------     ----    ----   -----     ----   -----   -------    ----   -----    ------
    1     STEP1         ( 1)         0        1      200.0    45.0     9.0      5.0     0.0    .9000       0       0    0.00000
    2     STEP1         ( 2)         0        2      200.0    45.0     9.0      5.0     0.0    .9000       0       0    0.00000
    3     STEP2         ( 1)         1        3      100.0    50.0    10.0      6.0     0.0    .8929       0       0    0.00000
    4     STEP2         ( 2)         1        4      100.0    50.0    10.0      6.0     0.0    .8929       0       0    0.00000
    5     STEP2         ( 3)         2        5      100.0    50.0    10.0      6.0     0.0    .8929       0       0    0.00000
    6     STEP2         ( 4)         2        6      100.0    50.0    10.0      6.0     0.0    .8929       0       0    0.00000
    7     CONTROL1      ( 1)         3      999        0.0    60.0    12.0      5.0     0.0    .9231       6       1    0.00000
    8     CONTROL2      ( 1)         4      999        0.0    60.0    12.0      5.0     0.0    .9231       6       3    0.00000
    9     CONTROL3      ( 1)         3      999        0.0    60.0    12.0      5.0     0.0    .9231       6       2    0.00000
   10     CONTROL4      ( 1)         5      999        0.0    60.0    12.0      5.0     0.0    .9231       6       4    0.00000
   11     STEP3         ( 1)         3        7      100.0    75.0    15.0     10.0     0.0    .8824      -5       0     .01500
   12     STEP3         ( 2)         4        8      100.0    75.0    15.0     10.0     0.0.   .8824      -5       0     .01500
   13     STEP3         ( 3)         5        9      100.0    75.0    15.0     10.0     0.0    .8824      -5       0     .01500
   14     STEP3         ( 4)         6       10      100.0    75.0    15.0     10.0     0.0    .8824      -5       0     .01500
   15     STEP4         ( 1)         7       11      100.0    60.0    12.0      7.0     0.0    .8955       1       0    0.00000
   16     STEP4         ( 2)         8       11      100.0    60.0    12.0      7.0     0.0    .8955       1       0    0.00000
   17     STEP4         ( 3)         9       11      100.0    60.0    12.0      7.0     0.0    .8955       1       0    0.00000
   18     STEP4         ( 4)        10       11      100.0    60.0    12.0      7.0     0.0    .8955       1       0    0.00000

GROUPED COMPONENTS
------------------

         GROUP    LAST ELEMENT    OUTPUT BUFFER
         -----    ------------    -------------
           1           11                7
           2           12                8
           3           13                9
           4           14               10
```

Figure 79 — Example 3.10 — Page 1

model is to be run for 120 intervals (600 minutes, or ten hours). Lastly, the buffer status table is to be printed every twelve intervals (one hour).

The first page of output does not completely reflect the complexity of this model. The 22 machine cards have been reduced to only 18 lines, i.e. each machine is listed only once. The AUX field of the Type 6 machines no longer contains the machine number of the associated Type -5 machine, but rather an internal sequencing of control devices. All Type 5 blocks constructed are listed, just as they are when only Type 5 machines are used. While this input listing appears rather simple, all the necessary links, both logical and physical, have been set up internally by GENMOD. That is, if a particular control device goes down, all the correct blocks will likewise be affected.

No machine handling table was printed for this run (NSPEC=0).

Because a print frequency of twelve was specified, along with IREW2=0, on the Execute Card, a full buffer status table was generated, containing current, average, maximum, and quartile breakdowns, as discussed earlier in para 3.6. In this example, each printout is meant to represent one hour of real time. Figure 80 contains the first six segments of buffer data for information purposes only. The next three segments (Intervals 84, 96, and 108) are deleted. The final segment (Interval 120) and the summary data are represented in Figure 81. Note that the projected average output is given both in terms of intervals and minutes and that the projection by shift length is also correct.

3.11 Remaining Points

This brings to a close the detailed construction and examination of models. Most of the features of GENMOD have been covered in depth.

Only component Types 4 and 99 have not actually appeared in an example. Type 4, though, acts very much like Type 2, except that the input is accepted in every interval, but released only once at the end of the cycle. Sufficient discussion was given of Type 99 (NO OP) to make their use clear.

While running a model with differing modes and purge/clear options was not done, these features were discussed.

The only print option not shown (NSPEC=2) simply turns off the input listing on the first page, so that if many runs are made of large models with no significant changes, much time and paper is saved.

It is hoped that the examples shown here have given the reader a thorough familiarization with the nature of a GENMOD simulation.

The prime purpose of GENMOD is to provide a quick look at a production design, the model of which can be embellished later. If your models reflect the simplicity with which GENMOD operates, you will have much success. Good luck.

```
TYPE 6 TEST CASE

PERIODIC BUFFER STATUS
----------------------

INTERVAL        1       2       3       4       5       6       7       8       9      10      11
--------------------------------------------------------------------------------------------------
  12          200.    200.    100.    100.    100.    100.     99.     99.     98.     99.   3546.
              200.    200.     92.     92.     92.     92.     82.     82.     82.     82.   1477.
             100/ 0  100/ 0  100/ 0  100/ 0  100/ 0  100/ 0  100/ 0  100/ 0  100/ 0  100/ 0  100/ 0
               0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/0     0/ 0
              200.    200.    100.    100.    100.    100.     99.     99.     99.     99.   3546.
--------------------------------------------------------------------------------------------------
  24          200.    200.    100.    100.    100.    100.     99.     98.     98.     98.   8274.
              200.    200.    100.    100.    100.    100.     99.     98.     98.     99.   6106.
             100/ 0  100/ 0  100/ 0  100/ 0  100/ 0  100/ 0  100/ 0  100/ 0  100/ 0  100/ 0  100/ 0
               0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0
              200.    200.    100.    100.    100.    100.     99.     99.     99.     99.   8274.
--------------------------------------------------------------------------------------------------
  36          200.      0.    100.    100.     68.     68.     99.     99.    265.     99.  12161.
              143.    178.     71.     71.     97.     97.     70.     70.    118.     99.  10464.
             100/ 0  100/ 0  100/ 0  100/0   100/ 0  100/ 0  100/ 0  100/ 0   97/ 3  100/ 0  100/ 0
               0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0
              200.    200.    100.    100.    100.    100.     99.     99.    265.     99.  12161.
--------------------------------------------------------------------------------------------------
  48          247.    313.     53.    354.    100.    241.     98.      0.    463.      0.  15671.
              204.    172.     96.    139.     72.    100.     98.     78.    417.     48.  13969.
             100/ 0   98/ 2  100/ 0   96/ 4  100/ 0  100/ 0  100/ 0  100/ 0   73/27  100/ 0  100/ 0
               0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0
              247.    313.    100.    354.    100.    241.     99.     99.    473.     99.  15671.
--------------------------------------------------------------------------------------------------
  60          560.   1000.    400.    808.    100.      0.    189.     99.     99.      0.  17943.
              380.    889.    232.    824.     50.    133.     83.     51.    197.     20.  16801.
              80/18   78/ 3   90/10   77/ 5  100/ 0  100/ 0  100/ 0  100/ 0   72/28  100/ 0  100/ 0
               2/ 0    2/17    0/ 0    3/15    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0
              560.   1000.    400.   1000.    100.    241.    189.    104.    461.     99.  17943.
--------------------------------------------------------------------------------------------------
  72          893.    320.    463.    686.    529.    550.    430.      0.      0.     47.  20908.
              879.    873.    463.    449.    195.    205.    408.     60.     63.     39.  19679.
              67/15   65/ 4   75/25   67/11   96/ 3   94/ 4   85/15  100/ 0   76/24  100/ 0  100/ 0
               3/15    4/26    0/ 0   10/12    1/ 0    1/ 0    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0
             1000.   1000.    463.    708.    529.    550.    442.     99.     99.     99.  20908.
--------------------------------------------------------------------------------------------------
```

Figure 80 — Example 3.10 — Page 4

```
          TYPE 6 TEST CASE

          PERIODIC BUFFER STATUS
          ----------------------

INTERVAL        1       2       3       4       5       6       7       8       9      10      11
---------------------------------------------------------------------------------------------------
  120         1000.   1000.   1000.   1000.    838.   1000.     98.      0.     98.      0.  31467.
              1000.    895.    837.   1000.    917.    999.     28.      0.     51.      0.  31164.
              40/ 9   66/ 4   58/32   40/ 7   57/ 2   57/24   72/28  100/ 0   86/14  100/ 0  100/ 0
               2/49    7/23   2/ 7     6/47   32/ 8    5/14    0/ 0    0/ 0    0/ 0    0/ 0    0/ 0
              1000.   1000.   1000.   1000.   1000.   1000.     99.      0.    196.      0.  31467.

MAXIMUM
MATERIAL
HANDLED-      1000.   1000.   1000.   1000.   1000.  1000.     442.    104.    473.    213.  31467.

PROJECTED AVERAGE OUTPUTS
-------------------------
         PER BASIC INTERVAL-          262.23 (   52.45 PER MINUTE
         PER 84.-INTERVAL SHIFT-    22027.
         PER 3/8/5 WEEK       -    330404.
         PER 63-SHIFT MONTH   -   1387698.
```

Figure 81 — Example 3.10 — Page 4 (Cont'd)

APPENDIX

DECK STRUCTURE

As described here, GENMOD exists as a FORTRAN program written for universal use. It was originally written for CDC equipment and at one time took advantage of some tricks available on those machines. However, when the first outside user asked for a version, the program was translated into IBM-compatible coding and has remained that way. More efficient coding could be used in many places by incorporating machine-dependent routines. Information about such versions is available upon request. At present, only one subroutine, which initializes the random number generator, is machine-dependent.

All examples given in this manual were run on a CDC-6600 system. Approximately 100,000 octal words of core storage are required. Running time, of course, depends upon the complexity of the model. A rough estimate, though, would be:

$$\text{CP seconds} = \frac{\text{Machines X Intervals}}{200}$$

Figure 82 gives a generalized picture of a complete run deck for a CDC-6600 system.

Also shown are sample suggested coding forms for both machine and buffer cards.

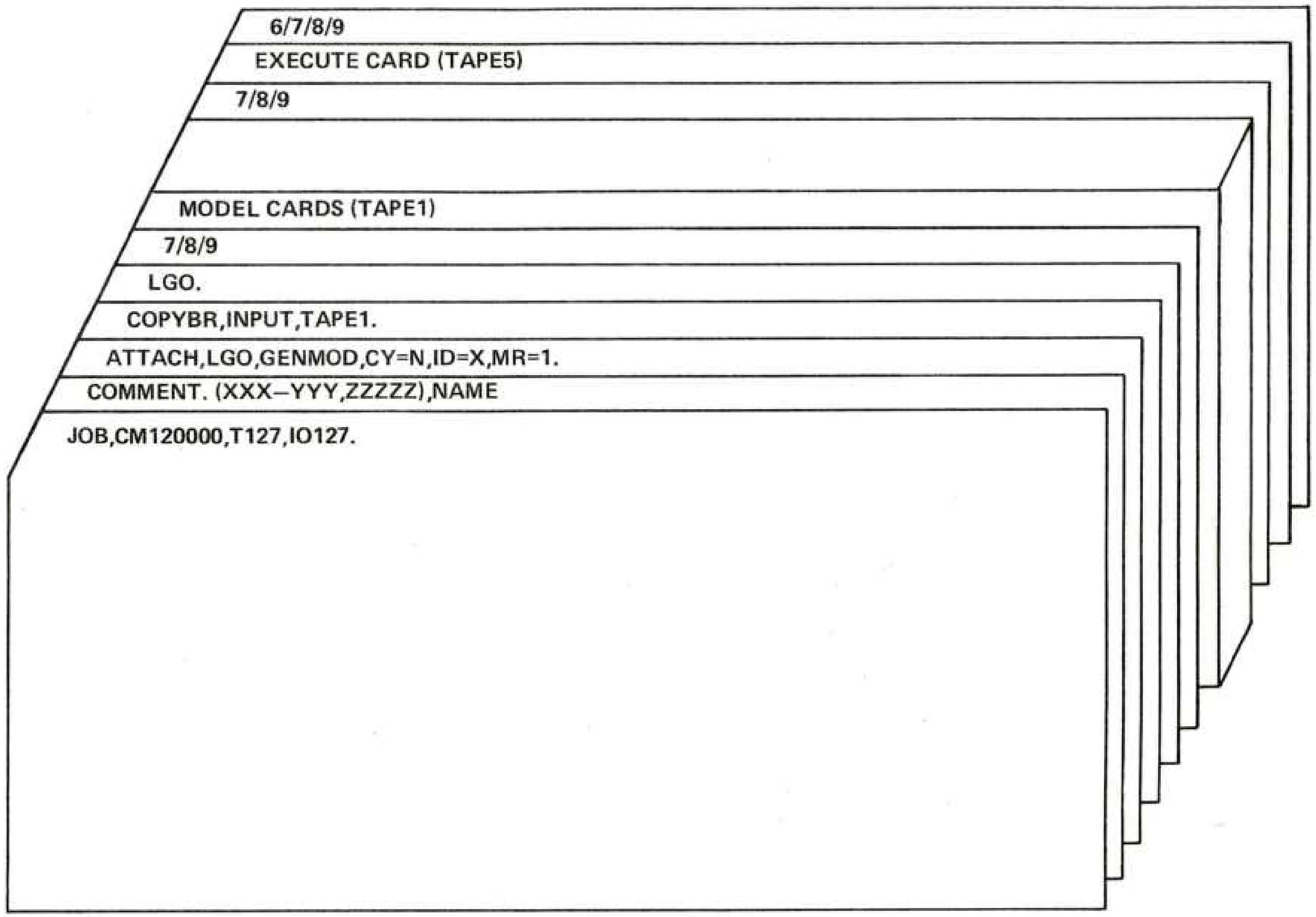


Figure 82 — CDC-6600 Deck Structure

GENMOD CODING SHEET

Title

1-80

Total Categories

1-3

Category Titles

1-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80

Total Machines

1-3

Starting Value

4-6

Machines

MACH	CAT	SEQ	IN	OUT	PPM	MTBF	STBF	MTTR	STTR	DEF	TYPE	AUX
1-3	4-6	7-9	10-12	13-15	16-23	24-31	32-39	40-47	48-55	56-63	64-66	67-69

GENMOD CODING SHEET

Machines

MACH	CAT	SEQ	IN	OUT	PPM	MTBF	STBF	MTTR	STTR	DEF	TYPE	AUX
1-3	4-6	7-9	10-12	13-15	16-23	24-31	32-39	40-47	48-55	56-63	64-66	67-69

GENMOD CODING SHEET

Total Buffers

1-3

Buffers

BUFF	SEED	MAX	PRINT	FEED
1-3	4-13	14-23	28	29-38

BUFF	SEED	MAX	PRINT	FEED
1-3	4-13	14-23	28	29-38

GENMOD CODING SHEET

Environment Card

RAWMAT	SHIFTL	IMODE	IDUR	IPURGE	ICLEAR	BIT
1-10	11-20	21-25	26-30	31-35	36-40	41-50

Execute Card

NSIMUL	NFREQ	NSPEC	IREW2	IREW3
1-5	6-10	11-15	16-20	21-25

This page intentionally left blank.

GLOSSARY

It is always best, when discussing a new subject, to agree on a definition of terms. This section contains a brief review of key words and phrases as they are used in this manual.

Alternator	— a machine which takes turns operating with its neighbors (Component Type 2)
Assembly	— the operation of matching parts from two or more input points and passing the result as a single unit to the output point (Type -3)
Batch	— an operation that requires more than one interval to perform; output is released in discreet chunks rather than continuously (Type 4)
Block	— a single indentifiable process step, characterized by one active component type
Branch	— a succession of machines/stations operating independently from other such arrangements
BTI	— Basic Time Interval — The real time equivalent of a single step through a GENMOD model
Buffer	— an inter-operation node through which parts must pass; may or may not have a real equivalent
Category	— an identifier assigned to a set of machines to allow statistics to be gathered at a level higher than individual components
Cell	— an internal shift register used in delaying delivery of parts; corresponds to a finite length of a moving conveyor
Competitive	— two or more machines, sharing a common input, which vie with each other for parts (Type 0)
Contingency	— a component which processes no parts, yet which must be in working condition in order for processing to continue (Type -5, Type 6)

GLOSSARY (Cont'd)

Defect	— a unit deterministically removed from a buffer in accordance with a specified probability range
Delay	— a process option by which parts are kept in transit between points for fixed periods of time
Domineering	— two or more machines, sharing a common input, which handle parts on a first-come-first-served basis (Type 1)
Downtime	— a count of intervals, at the start of which, a particular machine (or the entire line) was not in operating condition
Dummy Buffer	— a coding device designating an output point for a machine which does no real work; associated mainly with contingencies
Dummy Machine	— A Type 0 or Type 1 component which has been assigned an availability of 1.00 (i.e MTTR=0.0); acts as a perfect transferrer of parts to balance timing problems
Entry	— a buffer designated to receive uninterrupted flows of raw material; all models must have at least one entry (Buffer 0)
Environment	— a set of options invoked to describe the nature of the operation of a line as a whole, including mode and maintenance policies
Event	— an occurrence affecting the status of a machine, i.e. failure or repair; the time-to-next-event is continuously monitored by the program
Execute	— the process of actually carrying out a simulation from Interval 1 according to the aggregate contents of the model

GLOSSARY (Cont'd)

Failure	– a deterministic event by which a machine is inhibited from processing any material
Feedback	– the assignment of an earlier occurring buffer as the output point for a machine
Field	– a fixed set of columns on a punched card designated to contain a specific input parameter
Input	– the buffer point which supplies material to a machine or station; all GENMOD machines except Type 3 are allowed but one single input point; several machines, however, may draw from the same point
Inspection	– an operation designed to remove defective parts from a buffer; all active component types have the capability of doing so
Interval	– A single step through a GENMOD model during which all machines are given the opportunity to process parts and all such parts are moved between the appropriate buffers
Line	– a complete simulation model consisting of a set of linked operations and buffers, at least one entry point, and at least one final output point
Logbook	– a record of all failures and repairs experienced during a single execution of a model
Machine	– a single process step; same as block
Match	– the process of removing equal quantities from two or more input buffers and passing all as individuals to the output point (Type +3)
Mode	– the designated manner of model execution, i.e. continuous or with periodic pauses

GLOSSARY (Cont'd)

Model	– the set of machines, buffers, and environment to be simulated; same as line
Module	– a single part representing two or more smaller parts previously accounted for individually
MTBF	– Mean Time Between Failures – The statistical mean of the failure distribution assigned to a a particular machine
MTTR	– Mean Time to Repair – The statistical mean of the repair distribution assigned to a particular machine
Node	– an inter-operational point; same as buffer
NO-OP	– an operation which is completely ignored on a particular execution of a model; used to investigate design optimization (Type 99)
Operation	– a single process step; while designated precisely as one machine, it may logically consist of many machines or process steps
Packer	– an operation which removes fixed quantities of parts from a single buffer and passes them as one module to the output buffer
Parallel	– two or more machines which process parts that are logically similar, age wise
Part	– a single process unit for which accountability is maintained; can logically represent any convenient real equivalent
Probability Distribution	– the designated failure and repair patterns for a particular machine; failures may take on either normal or exponential patterns; repairs may be either log-normal or exponential

GLOSSARY (Cont'd)

Proportional Splitter	– two or more machines fed by a common input buffer which distribute parts according to fixed ratios, rather than competitively or cooperatively
Quartile	– an internally maintained sub-division of a buffer equal to one-quarter of its capacity; statistics are gathered on the demand for one, two, three or all four of such quarters
Raw Material	– the parts fed to an entry point each interval; the flow is considered constant
Repair	– a deterministic event by which a machine is restored to operating condition following a failure
Residence time	– the sum of the intervals required to travel from the earliest entry point of a model to the furthest output point, including delay times and batch durations
Sample	– the statistical process by which a uniform random deviate is equated to a real world equivalent by means of a cumulative probability distribution
Seed	– a quantity of parts placed in a buffer prior to execution of a model
Series	– two or more operations which follow each other directly either physically or logically
Shift	– a designated duration of simulation equated to a convenient real production run, e.g. one day
Station	– two or more machines drawing from a common input point and feeding a common output point
Status	– the operating condition of a machine at the start of an interval; each machine is either available (up) or not available (down)

GLOSSARY (Cont'd)

Steady state	— a period of production characterized by a wide distribution of parts through all buffers and rhythmic failure/repair patterns of all machines
Stop time	— a count of intervals during which a machine was forced to stop due to an empty input buffer or slow due to a full output buffer

INDEX TO PARAGRAPH HEADINGS

INDEX TO PARAGRAPH HEADINGS (Cont'd)

INDEX

Page

DISTRIBUTION LIST

Defense Documentation Center (12 copies)
Cameron Station
Alexandria, VA 22314

Commander
U.S. Army Armament Reasearch and Development Command (5 copies)
ATTN: DRDAR-TSS
Dover, NJ 07801

Weapon System Concept Team/CSL
ATTN: DRDAR-ACW
Aberdeen Proving Ground, MD 21010

Technical Library
ATTN: DRDAR-CLJ-L
Aberdeen Proving Ground, MD 21010

Technical Library
ATTN: DRDAR-TSB-S
Aberdeen Proving Ground, MD 21005

Benet Weapons Laboratory
Technical Library
ATTN: DRDAR-LCB-TL
Watervliet, NY 12189

Commander
U.S. Army Armament Materiel Readiness Command
ATTN: DRSAR-LEP-L
Rock Island, IL 61299

Director
U.S. Army TRADOC Systems Analysis Activity
ATTN: ATAA-SL (Tech Lib)
White Sands Missile Range, NM 88002